

(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有權機關  
國際事務局



(43) 國際公開日  
2004 年 3 月 4 日 (04.03.2004)

**PCT**

(10) 国際公開番号  
**WO 2004/019553 A1**

**(51) 國際特許分類<sup>7</sup>:**

**H04L 9/32**

(71) 出願人(米国を除く全ての指定国について): 株式会社  
エイシーエス (ADVANCED COMPUTER SYSTEMS,  
INC.) [JP/JP]; 〒211-0005 神奈川県 川崎市 中原区新  
丸子町915-15 Kanagawa (JP). 株式会社産学連携機構  
九州 (KYUSHU TLO COMPANY, LIMITED) [JP/JP];  
〒812-8581 福岡県 福岡市 東区箱崎6丁目10番1号  
Fukuoka (JP).

(21) 國際出願番号:

PCT/JP2003/007794

(22) 國際出願日:

2003 年6 月19 日 (19.06.2003)

(25) 国際出願の言語:

日本語

(26) 国際公開の言語:

日本語

(30) 優先権データ:

特願2002-178947 2002年6月19日(19.06.2002)

特願2003-69375 2003年3月14日 (14.03.2003)

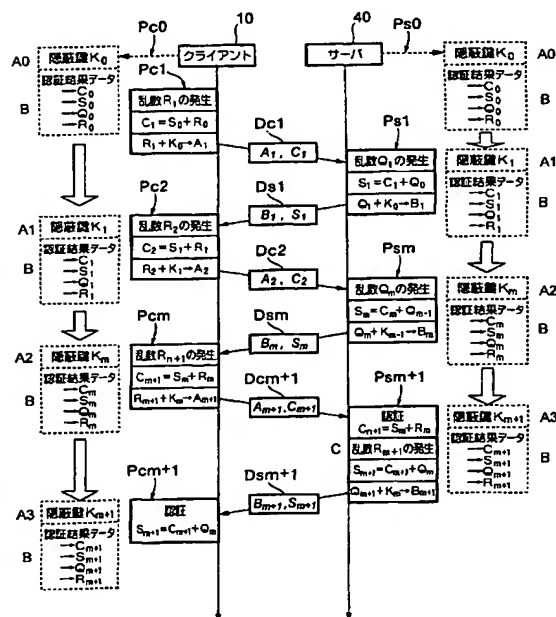
(72) 発明者; および

(75) 発明者/出願人 (米国についてのみ): 今本 健二 (IMAMOTO, Kenji) [JP/JP]; 〒812-0061 福岡県福岡市東区筥松4-22-14 松原寮B-313 Fukuoka (JP). 大河

〔統葉有〕

**(54) Title: INTER-AUTHENTICATION METHOD AND DEVICE**

(54) 発明の名称: 相互認証方法及び装置



**(S7) Abstract:** An inter-authentication method capable of safely and easily performing inter-authentication. In the inter-authentication process, a private key  $K_0$  of the initial value is stored in a client and a server ( $Pc0$ ,  $Ps0$ ). The client generates a random number  $R$ , calculates password data  $C$  and authentication data  $A$ , and transmits the result to the server ( $Pc1$ ). The server receives the authentication data  $A$  and the password data  $C$  from the client, generates a random number  $R$ , calculates and returns password data  $S$  and authentication data  $Q$ , and updates the private key  $K_0$  to a new private key  $K_1$  ( $Ps1$ ). The client receives the authentication data  $B$  and the password data  $S$  from the server, generates a random number  $R$ , calculates the password data  $C_2$  and the authentication data  $A_2$ , returns the results to the server, and updates the private key  $K_0$  to the new private key  $K_1$  ( $Pc2$ ). The client and the server check whether validity is satisfied.

A0...PRIVATE KEY  $K_0$   
 B...AUTHENTICATION RESULT DATA  
 A1...PRIVATE KEY  $K_1$   
 A2...PRIVATE KEY  $K_m$   
 A3...PRIVATE KEY  $K_{m+1}$   
 10...CLIENT  
 Pc 1...GENERATION OF RANDOM NUMBER  $R_1$   
 Pc 2...GENERATION OF RANDOM NUMBER  $R_2$   
 Pcm ...GENERATION OF RANDOM NUMBER  $R_{n+1}$   
 Pcm +1...AUTHENTICATION  
 40...SERVER  
 Pa 1...GENERATION OF RANDOM NUMBER  $Q_1$   
 Psm ...GENERATION OF RANDOM NUMBER  $Q_m$   
 Psm +1...AUTHENTICATION  
 C...GENERATION OF RANDOM NUMBER  $Q_{m+1}$

〔統葉有〕



克好 (OKAWA, Katsuyoshi) [JP/JP]; 〒195-0055 東京都町田市 三輪緑山1-3-2 緑山ヒルズ212 Tokyo (JP). 橋本努 (HASHIMOTO, Tsutomu) [JP/JP]; 〒184-0002 東京都 小金井市 梶野町5-2-27-206 Tokyo (JP).

(74) 代理人: 酒井 一, 外 (SAKAI, Hajime et al.); 〒102-0083 東京都 千代田区 麹町5丁目7番地 秀和紀尾井町 T B R ビル Tokyo (JP).

(81) 指定国 (国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK,

SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) 指定国 (広域): ARIPO 特許 (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア特許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許 (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告書

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

#### (57) 要約:

安全かつ簡便に相互認証することができる相互認証方法を得る。そのために、相互認証プロセスにおいて、クライアント及びサーバに初期値の隠蔽鍵  $K_0$  を格納する ( $P_{c0}$ ,  $P_{s0}$ )。クライアントは乱数  $R$  を生成し暗証データ  $C$  及び認証データ  $A$  を計算し、サーバに送信する ( $P_{c1}$ )。サーバはクライアントから認証データ  $A$  及び暗証データ  $C$  を受信しかつ乱数  $R$  を生成し暗証データ  $S$ 、認証データ  $Q$  を計算し返信し、隠蔽鍵  $K_0$  を新規の隠蔽鍵  $K_1$  に更新する ( $P_{s1}$ )。クライアントはサーバから認証データ  $B$  及び暗証データ  $S$  を受信しかつ乱数  $R$  を生成し暗証データ  $C_2$ 、認証データ  $A_2$  を計算しサーバに返信し、隠蔽鍵  $K_0$  を新規の隠蔽鍵  $K_1$  に更新する ( $P_{c2}$ )。クライアント及びサーバは正当性が成立するか検査する。

## 明 細 書

## 相互認証方法及び装置

## 技術分野

本発明は、ネットワークに接続されたコンピュータシステム等の装置における相互認証方法及び装置並びにこれに用いるワンタイムIDの生成方法、認証方法、認証システム、サーバ、クライアントおよびプログラムに関するものである。

とりわけ、本発明は、少なくとも第1認証装置と第2認証装置との間の関係についての正当性を検証する相互認証方法及び装置、並びに複数の装置間またはアプリケーション間における認証に用いて好適なワンタイムIDの生成方法、上記ワンタイムIDを用いた認証方法、認証システム、サーバ、クライアントおよびプログラムに関するものである。

ネットワーク上において、あるユーザが自分の身元を証明するためには、認証が必要である。認証とは、何らかのプロトコルにより、証明者が自分の身元を検証者に対して証明することであり、電子商取引などの分野において、必須の技術である。例えば、ユーザがサーバに対して、身元を証明したいときは、ユーザが証明者に、サーバが検証者に対応する。また、逆に、サーバがユーザに対して身元を証明したい場合は、サーバが証明者に、ユーザが検証者に対応する。1対1の装置の間では、その立場が反転する場合があるので、相互に認証する相互認証が必要である。

相互認証は、ユーザとサーバの間に限定されず、任意のコンピュータ間での身元を証明する方法として、幅広く利用されている。最近では、公開鍵暗号を用いたものが知られており、証明者は公開鍵と秘密鍵を所持しており、証明者が公開鍵に対応する秘密鍵を所持することを、なんらかのプロトコルにより、検証者に示すことにより、身元を証明している。

しかしながら、従来の相互認証方法では、認証に用いる鍵が単一であるため、ひとたび鍵が知られてしまうと、第三者がユーザになりすまして認証される場合がある。また、ユーザは、鍵の保管に注意を払わなければならない、簡便に利用す

ることができなかった。

例えば、インターネットのような非同期型ネットワークでは、複数のコンピュータが同時に通信しており、証明者が複数の検証者と同時にプロトコルを実行する場合がある。WWW (World Wide Web: ワールド・ワイド・ウェブ) では、HTTP (Hypertext Transfer Protocol: WWWサーバとWWWブラウザやWebブラウザ等が、ファイル等の情報授受に使うプロトコル) のサーバと、接続先であるクライアントとの間では、多数の認証が要求される。

また、上記のような相互認証技術において、従来より、ネットワークを介してコンピュータ間 (例えば、クライアント・サーバ間) で通信を行う際には、不正なアクセス等を排除するために、サービス等の提供に先立って認証が行われる。この認証においては、第三者が知り得ない所定の秘密情報 (例えば、IDやパスワード、乱数、或いはそれら情報を引数とする関数値など) を予め双方が共有し、その秘密情報に基づいて各々の正当性を相互に検証するのが一般的である。

他方、IETF (Internet Engineering Task Force) が公式に発行するRFC (Request For Comments) においては、インターネットでIPパケットの暗号化と認証を行なうセキュリティプロトコルとして、IPsec (Security Architecture for Internet Protocol) が規定されている。このIPsecでは、暗号・認証のパラメータを動的に生成して交換するIKE (Internet Key Exchange) という自動鍵交換のプロトコルが標準的に採用されている (例えば、特開2002-374238号公報 (段落番号0002~0009) 参照)。

そして、近年では、このIKEの方式にワンタイムIDを導入して、既知共有鍵を用いたIKEの方式で問題となっていた、ID情報保護、DoS (Denial of Service attack) 攻撃防止、リモートアクセスなどを実現したP-SIGMAと呼ばれる鍵交換・認証方式が提案されている。

このP-SIGMAにおいては、例えば、図1に示すような手順で鍵交換および認証が行われている。

先ず、クライアントが、SA (Security Association) の提案、乱数 $R_c$ 、D



H (Diffie-Hellman) 公開値  $g^x$ 、OID (ワンタイム ID) をサーバに対して送信する。なお、SA の提案には、暗号アルゴリズムや認証方式、鍵交換に使用するパラメータ等に関する提案が含まれている。

次いで、サーバが、受け取った OID からクライアントを識別し、識別できない場合には、通信を拒否する。識別できる場合には、受諾した SA、乱数  $R_s$ 、DH 公開値  $g^y$ 、HASH  $s$ 、セッション鍵  $e$  で暗号化した ID  $s$  (サーバ ID) をクライアントに対して送信する。なお、セッション鍵  $e$  は、既知共有鍵、乱数  $R_s$ 、乱数  $R_c$  および DH 共通鍵  $g^{xy}$  を引数とする鍵付きハッシュ関数の関数値であり、HASH  $s$  は、既知共有鍵、乱数  $R_s$ 、乱数  $R_c$ 、DH 公開値  $g^x$ 、 $g^y$  および ID  $s$  を引数とする疑似乱数関数の関数値である。

次いで、クライアントが、受け取った HASH  $s$  を検証し、この HASH  $s$  に基づいてサーバの正当性を確認する。HASH  $s$  が正しければ、HASH  $c$ 、セッション鍵  $e$  で暗号化した ID  $c$  (クライアント ID) をサーバに対して送信する。ここで、HASH  $c$  は、既知共有鍵、乱数  $R_s$ 、乱数  $R_c$ 、DH 公開値  $g^x$ 、 $g^y$  および ID  $c$  を引数とする疑似乱数関数の関数値である。

次いで、サーバが、受け取った HASH  $c$  を検証し、この HASH  $c$  に基づいてクライアントの正当性を確認する。HASH  $c$  が正しければ、当該プロトコルを終了する。

この P-SIGMA において、OID (ワンタイム ID) は、次のように定義されている。

$$OID1 = \text{prf}(K, 1)$$

$$OID2 = \text{prf}(K, 2)$$

...

$$OIDn = \text{prf}(K, n)$$

..... (式 1)

この定義式において、 $OIDn$  は  $n$  番目の SA 確立時に用いられるワンタイム ID、 $\text{prf}$  は疑似乱数関数、 $K$  は既知共有鍵、若しくは既知共有鍵から生成された値である。

このため、上記P-SIGMAによれば、OIDを導入したことにより、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればOIDを識別情報として把握できるといった効果が得られるとともに、クライアント・サーバ間で通信が行われる度（すなわち、SAの生成または更新毎）にOIDが変更されるため、第三者が次回のOIDを予測することができないといった効果が得られる。

しかしながら、上記P-SIGMAにおいては、既知共有鍵がひとたび知られてしまうと、すべてのOIDが予測されてしまい、その結果、OIDの将来にわたる安全性（すなわち、PFS: Perfect Forward Security）を保証できなくなるといった問題点があった。

以上、具体例として、P-SIGMAと呼ばれる鍵交換・認証方式について述べてきたが、一般に、ワンタイムIDを用いて複数の装置間またはアプリケーション間における認証を行う認証方式では、特定の秘密情報に基づいてすべてのワンタイムIDの生成を行っており、上記同様の問題点を有している。

本発明は、上記従来技術における種々の不具合事実を考慮してなされたもので、その第1の目的は、安全かつ簡便に相互認証することができる相互認証方法及び装置を得ることである。

本発明の第2の目的は、盗聴が困難で安全性に優れたワンタイムIDの生成方法、上記ワンタイムIDを用いた認証方法、認証システム、サーバ、クライアントおよびプログラムを提供することである。

## 発明の開示

上記目的を達成するために本発明は、通信回線を介して接続された第1認証装置と第2認証装置の相互関係を認証する相互認証方法であって、前記第1認証装置を特定するための記憶データと、第2認証装置を特定するための記憶データとを、前記第1認証装置及び第2認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを用いて更新した更新結果を履歴データとして、前記第1認証装置及び第2認証装置の各々に共通に記憶する記憶工程を含む

ものである。前記第 1 認証装置は、記憶されている履歴データを用いて記憶データを新規に生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第 2 認証装置に送信する第 1 送信工程と、前記第 2 認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第 1 更新工程とを含み、また、前記第 2 認証装置は、前記第 1 認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第 1 認証装置に送信する第 2 送信工程と、前記第 1 認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第 2 更新工程とを含む。そして、前記第 1 認証装置及び第 2 認証装置の少なくとも一方の装置において、履歴データに基づいて記憶データの正当性が成立したときに、第 1 認証装置と第 2 認証装置の相互関係が正当であると検証する。

本発明はまた、上記相互認証方法を実施するための相互認証装置を実現するものである。この相互認証装置は、通信回線を介して接続された第 1 認証装置と第 2 認証装置とから成り、前記第 1 認証装置と第 2 認証装置との間の相互関係を認証するものであって、前記第 1 認証装置に設けられ当該第 1 認証装置を特定するための記憶データを格納する第 1 のメモリと、前記第 2 認証装置に設けられ当該第 2 認証装置を特定するための記憶データを格納する第 2 のメモリと、前記第 1 認証装置及び第 2 認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを格納する認証データ記憶手段と、前記認証データを用いて更新した更新結果を履歴データとして、前記第 1 認証装置及び第 2 認証装置の各々に共通に記憶する履歴データ記憶手段と、前記第 1 認証装置又は第 2 認証装置のうち、認証用データ送信側の認証装置に設けられ、前記履歴データを用いて記憶データを新規に生成する記憶データ生成手段と、生成した新規の記憶データを前記履歴データを用いて暗号化して認証用データ受信側の認証装置に送信する第 1 の送信手段と、認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成する記憶データ生成手段と、生成した新規の記憶データを

前記履歴データを用いて暗号化して前記認証用データ送信側の認証装置に返信する第2の送信手段と、認証用データ送信側の認証装置に設けられ、前記認証用データ受信側の認証装置から返信された記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第1の更新手段と、認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び前記返信した新規の記憶データによって前記履歴データを更新する第2の更新手段とを含み、前記第1認証装置及び第2認証装置の少なくとも一方の装置において、前記履歴データに基づいて記憶データの正当性が成立したときに、第1認証装置と第2認証装置の相互関係が正当であると検証する検証手段とを備えている。

かかる相互認証装置は、生成した新規の記憶データを前記履歴データを用いて暗号化するための認証用データを演算する演算手段を有することができる。また、この相互認証装置は、前記演算手段により認証用データを生成するに際し、暗号化用のデータを生成する乱数発生手段を有することも可能である。

また、本発明では、第1認証装置を特定するための記憶データと、第2認証装置を特定するための記憶データとを、履歴データとして、前記第1認証装置及び第2認証装置の各々に共通に記憶する。この履歴データは、第1認証装置及び第2認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを用いて更新した更新結果である。第1認証装置は、記憶されている履歴データを用いて新規の記憶データを生成しかつ新規の記憶データを記憶されている履歴データを用いて暗号化して第2認証装置に送信する。これを第2認証装置が受け取り、第2認証装置は第1認証装置からの記憶データ及び記憶されている履歴データを用いて新規の記憶データを生成しかつ新規の記憶データを記憶されている履歴データを用いて暗号化して第1認証装置に送信する。このとき、第1認証装置は、第2認証装置からの記憶データ及び送信した新規の記憶データによって履歴データを更新する。また、第2認証装置は、第1認証装置からの記憶データ及び送信した新規の記憶データによって履歴データを更新する。この送信工程の後には、第1認証装置及び第2認証装置の少なくとも一方の装置において、履

履歴データに基づいて記憶データの正当性が成立したときに、第1認証装置と第2認証装置の相互関係が正当であると検証する。すなわち、第1認証装置及び第2認証装置の一方の認証装置では、他方の認証装置からの履歴を含むデータを受け取り、記憶されている履歴データと照合することが可能となる。そして送信するときには、記憶されている履歴データから新規で異なる履歴データに基づくデータを送信するので、同一データによる授受はない。このため、秘匿性を向上させることもできる。

より詳細には、前記履歴データを履歴データKとして、該履歴データKとして記憶する、前記第1認証装置を特定するための記憶データは、暗証データC及び認証データRであり、前記第2認証装置を特定するための記憶データは、暗証データS及び認証データQであることを特徴とする。

前記第1送信工程は、記憶されている履歴データKの暗証データS及び認証データRを用いて暗証データCを新規に生成しかつ、記憶されている履歴データKの認証データRについて新規に生成し、生成した新規の認証データRを前記履歴データKを用いて暗号化して認証データAを求め、前記認証データA及び新規の暗証データCを第2認証装置に送信し、前記第1更新工程は、前記第2認証装置からのデータを受信し、前記送信した新規の暗証データC、受信した新規に生成された暗証データS、受信した新規に生成された認証データQ、及び前記送信した新規の認証データRにより、前記履歴データKを更新し、前記第2送信工程は、前記第1認証装置からのデータを受信し、受信した新規の暗証データC及び記憶されている履歴データKの認証データQを用いて暗証データSを新規に生成しかつ記憶されている履歴データKの認証データQについて新規に生成し、生成した新規の認証データQを記憶した履歴データKを用いて暗号化して認証データBを求め、前記認証データB及び新規の暗証データSを第1認証装置に送信し、前記第2更新工程は、受信した新規の暗証データC、新規に生成した暗証データS、新規に生成した認証データQ、及び受信した新規の認証データRにより、前記履歴データKを更新し、前記第1認証装置及び第2認証装置の少なくとも一方の装置において、履歴データKに基づいて暗証データの正当性が成立したときに、第

1 認証装置と第 2 認証装置の相互関係が正当であると検証することを特徴とする。

前記記憶工程は、前記第 1 送信工程、第 1 更新工程、第 2 送信工程、及び第 2 更新工程における認証による更新結果を履歴データとして記憶することを特徴とする。

前記認証データ R 及び認証データ Q の少なくとも一方は、乱数発生手段により発生された乱数、データ容量、時間データの少なくとも 1 つであることを特徴とする。

前記第 1 認証装置の第 1 送信工程では、前記暗証データ S 及び認証データ R による予め定めた関数の演算結果の値を暗証データ C として生成し、前記第 2 認証装置の第 2 送信工程では、前記暗証データ C 及び前記認証データ Q による予め定めた関数の演算結果の値を暗証データ S として生成することを特徴とする。

前記第 1 認証装置の第 1 送信工程では、前記生成した新規の認証データ R 及び前記履歴データ K による予め定めた関数の演算結果の値を認証データ A として求め、前記第 2 認証装置の第 2 送信工程では、前記生成した新規の認証データ Q 及び前記履歴データ K による予め定めた関数の演算結果の値を認証データ B として求めることを特徴とする。

前記第 1 認証装置の検証工程は、前記履歴データ K のうち記憶されている認証データ Q 及び前回送信する前に生成した暗証データ C による予め定めた関数の演算結果の値が受信した暗証データ S と一致するときに前記相互関係が正当であると検証することを特徴とする。

前記第 2 認証装置の検証工程は、前記履歴データ K のうち記憶されている暗証データ S 及び認証データ R による予め定めた関数の演算結果の値が受信した暗証データ C と一致するときに前記相互関係が正当であると検証することを特徴とする。

前記記憶工程は、前記第 1 送信工程、第 2 送信工程、第 1 更新工程及び第 2 更新工程を複数実施した結果、得られるデータを履歴データ K として記憶することを特徴とする。

以上の説明から明らかなように、本発明によれば、第 1 認証装置及び第 2 認証

装置の間で相互認証するときに、第1認証装置及び第2認証装置の各々に共通に履歴データを記憶すると共に、履歴データを更新するので、安全かつ簡便に相互認証することができ、例えば、クライアント・コンピュータとサーバ・コンピュータとの間で授受される情報から、クライアント・コンピュータの鍵が漏洩することがなく、確実に認証が行えるという効果がある。

本発明はまた、上記相互認証方法及び装置において使用されるワンタイムIDの生成方法を提供する。これは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

ここで、一方向関数とは、引数から結果（関数値）を求めるのは簡単であるが、結果から引数を求めるのは難しい関数のことを云い、この一方向関数には、例えば、ハッシュ関数、疑似乱数関数などが含まれる。

所定の通信単位としては、例えば、IPsecにおいてSAが確立されてから当該SAが無効になるまでの間にクライアント・サーバ間で行われる一連の通信を、所定の通信単位として設定することも可能であるし、装置間またはアプリケーション間で行われる1回のデータ送受信を所定の通信単位として設定することも可能である。

可変共有鍵は、上記所定の通信単位毎に変化し、且つ認証を行う装置間またはアプリケーション間で共有される、第三者が知り得ない秘密情報であれば、如何なる鍵であってもよい。

認証とは、一方の装置（または一方のアプリケーション）が他方の装置（または他方のアプリケーション）にアクセスする際に、他方の装置が一方の装置の正当性を確認することを云い、識別情報とは、上記認証において少なくとも一方の装置から他方の装置に送信されて当該他方の装置が一方の装置を識別するのに用

いる情報（ID）のことを云う。

また、上記認証には、一方の装置が他方の装置の認証を行う一方向認証と、双方の装置で相互に認証を行う相互認証とが含まれる。例えば、上記認証においてワンタイムIDを使用する方法としては、双方の装置でワンタイムIDを生成するとともに、一方の装置が他方の装置にワンタイムIDを送信し、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、一方の装置を識別或いは認証する方法が挙げられる。

本発明は、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

本発明はまた、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

本発明はまた、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、



この関数値から上記ワнтаイムIDを生成するようにしたことを特徴とするものである。

本発明はまた、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワнтаイムIDとして、当該ワнтаイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワнтаイムIDを送信して、他方の装置が、一方の装置から受信したワнтаイムIDと自らが生成したワнтаイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワнтаイムIDを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワнтаイムIDを生成するようにしたことを特徴とするものである。

本発明はまた、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワнтаイムIDとして、当該ワнтаイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワнтаイムIDを送信して、他方の装置が、一方の装置から受信したワнтаイムIDと自らが生成したワнтаイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワнтаイムIDを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワнтаイムIDを生成するようにしたことを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成し、このワнтаイムID（S I G N A L<sub>n</sub>）を用いて、互いに通信を行なう第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、上記第二装置との間で予め共有化された可変共有鍵を用いて上記ワнтаイムIDを生成するとともに、この生成したワнтаイムIDと、当該第一装置に予め設定さ

れたIDを少なくとも引数とする一方向関数 $F_c$ の関数値と、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方とを上記第二装置に対して送信するステップと、上記第二装置が、上記ワンタイムIDおよび上記一方向関数 $F_c$ の関数値を演算により求め、この演算結果と、上記第一装置から受信したワンタイムIDおよび一方向関数 $F_c$ の関数値との照合により、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、当該第二装置に予め設定されたIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とを上記第一装置に対して送信するステップと、上記第一装置が、上記一方向関数 $F_s$ の関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数 $F_s$ の関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記一方向関数 $F_c$ として、所定の共有鍵、上記Diffie-Hellman公開値の一方、上記第一装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いるとともに、上記一方向関数 $F_s$ として、上記所定の共有鍵、上記Diffie-Hellman公開値の一方、上記Diffie-Hellman公開値の他方、上記第二装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いることを特徴とするものである。

本発明はまた、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、上記第二装置との間で予め共有化された第一の可変共有鍵と当該第一装置の通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムID (SIGNAL<sub>n,j</sub>) として生成するとともに、上記第一の可変共有鍵を用いて、当該第一装置に予め設定されたID、上記第二装置に予め設定されたID、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムIDを暗号化し、この暗号化データと上記第一のワンタイムIDとを上記第二装置に対して送信するステップと、上

記第二装置が、上記第一のワンタイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムIDとの照合により、上記第一装置を識別するステップと、上記第二装置が、上記第一装置を識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第一のワンタイムIDに基づいて、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、上記第一の可変共有鍵と当該第二装置の通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムID (SIGNAL' n,1) として生成するとともに、上記第一装置から受信したDiffie-Hellman公開値の一方と当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二のワンタイムIDを引数とする一方向関数hの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記第一装置に対して送信するステップと、上記第一装置が、上記第二のワンタイムIDを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムIDとの照合により、上記第二装置を識別するステップと、上記第一装置が、上記第二装置を識別できた場合に、上記第二装置から受信した上記Diffie-Hellman公開値の他方と当該第一装置に予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数hの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数hの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記第二のワンタイムIDを生成する一方向関数として、上記第一のワンタイムIDを生成する一方向関数とは異なる一方向関数を用いるようにしたことを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、

所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いて、第一装置と第二装置間における認証（相互認証）を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムID（ $SIGNAL_{c1}$ ）として求め、この第一のワンタイムIDと上記第一の乱数とを上記第二装置に対して送信するステップと、上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムID（ $SIGNAL_{s1}$ ）として求め、この第二のワンタイムIDと上記第二の乱数とを上記第一装置に対して送信するステップと、上記第一装置が、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムIDとの比較により、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムID（ $SIGNAL_{c2}$ ）として求め、この第三のワンタイムIDを上記第二装置に対して送信するステップと、上記第二装置が、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いて、第一装置と第二装置間における認証（相互認証）を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置

との間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワнтаイムID (SIGNAL<sub>c1</sub>) として求め、この第一のワнтаイムIDと上記第一の乱数を上記第二装置に対して送信するステップと、上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムID (SIGNAL<sub>s1</sub>) として求め、この第二のワнтаイムIDと上記第二の乱数を上記第一装置に対して送信するステップと、上記第一装置が、上記第一の乱数および上記共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記第二装置から受信した上記第二のワнтаイムIDとの比較により、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワнтаイムID (SIGNAL<sub>c2</sub>) として求め、この第三のワнтаイムIDを上記第二装置に対して送信するステップと、上記第二装置が、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワнтаイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワнтаイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とするものである。

本発明はまた、上記認証方法において、上記第二装置が上記第二のワнтаイムIDと上記第二の乱数とを上記第一装置に対して送信するステップにおいて、上記第二装置は、上記第一装置との間で予め共有化された乱数を初期乱数として、この初期乱数と上記第一の乱数を引数とする所定の演算を行い、この演算結果を上記第一装置に対して送信する一方、上記第一装置は、上記第二装置の正当性の判定材料として、上記第二装置から受信した上記演算結果を、上記第二のワнтаイムIDとともに用いることを特徴とするものである。

本発明はまた、上記認証方法において、上記第一装置が上記第三のワнтаイムIDを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記

第一の乱数と上記第二の乱数を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワンタイムIDとともに用いることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムID (S I G N A L<sub>ci</sub>) として求め、当該第一装置に予め設定されたID、上記第二装置に予め設定されたIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムIDとを上記第二装置に対して送信するステップと、上記第二装置が、上記第一のワンタイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムIDとの照合により、上記第一装置を識別するステップと、上記第二装置が、上記第一装置を識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたIDおよび当該第二装置に予め設定されたIDに基づいて、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムID (S I G N A L<sub>si</sub>) として求め、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムIDとを上記第一装置に対して送信するステップと、上記第二装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップと、上記第一装置が、上記第二のワンタイムIDを演算により求め、この演算結果と、上

記第二装置から受信した上記第二のワンタイムIDとの照合により、上記第二装置を識別するステップと、上記第一装置が、上記第二装置を識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる、上記第二装置に予め設定されたIDおよび当該第一装置に予め設定されたIDに基づいて、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップとを有することを特徴とするものである。

本発明はまた、上記認証方法において、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後に、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようにしたことを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムID (S I G N A L<sub>n</sub>) を用いてクライアントとの間で認証を行うサーバであって、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数F<sub>c</sub>の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムIDとを上記クライアントから受信する受信手段と、上記一方向関数の関数値F<sub>c</sub>および上記ワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムIDおよび上記一方向関数F<sub>c</sub>の関数値との比較により、上記クライアントの正当性を判定する判定手段と、上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数F<sub>s</sub>の関数値と、当該サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する送信手段とを備えることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムID (S I G N A L<sub>n</sub>) を用いてサーバとの間で認証を行

クライアントであって、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワнтаイムIDを生成するとともに、当該クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値を演算により求め、これらワнтаイムIDおよび一方向関数 $F_c$ の関数値と、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する送信手段と、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、上記一方向関数 $F_s$ の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数 $F_s$ の関数値との比較により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして、上記サーバと、上記クライアントとを備えてなることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成し、このワнтаイムID (SIGNAL<sub>n</sub>) に基づいてクライアントとの間で認証を行うサーバに実行させるプログラムであって、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワнтаイムIDとを上記クライアントから受信する処理と、上記一方向関数の関数値 $F_c$ および上記ワнтаイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記ワнтаイムIDおよび上記一方向関数 $F_c$ の関数値との比較により、上記クライアントの正当性を判定する処理と、上記クライアントが正当であると判定された場合に、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する処理とを上記サーバに実行させることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有



鍵を引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成し、このワнтаイムID (S I G N A L<sub>n</sub>) に基づいてサーバとの間で認証を行うクライアントに実行させるプログラムであって、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワнтаイムIDを生成するとともに、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数F<sub>c</sub>の関数値を演算により求め、これらワнтаイムIDおよび一方向関数F<sub>c</sub>の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する処理と、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数F<sub>s</sub>の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する処理と、上記一方向関数F<sub>s</sub>の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数F<sub>s</sub>の関数値との比較により、上記サーバの正当性を判定する処理とを上記クライアントに実行させることを特徴とするものである。

本発明はまた、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成し、このワнтаイムIDを用いてクライアントとの間で認証を行うサーバであって、上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワнтаイムID (S I G N A L<sub>n,j</sub>) として、この第一のワнтаイムID、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバID、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワнтаイムIDとを上記クライアントから受信する受信手段と、上記第一のワнтаイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワнтаイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントID、上記サーバIDおよび上記第一のワнтаイムIDに基づいて、上記クライアントの正当性を判定する判定

手段と、上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワнтаイムID (SIGNAL' n, i) として生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントID、上記サーバIDおよび上記第二のワнтаイムIDを引数とする一方向関数hの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワнтаイムIDとを上記クライアントに対して送信する送信手段とを備えることを特徴とするものである。

本発明はまた、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成し、このワнтаイムIDを用いてサーバとの間で認証を行うクライアントであって、上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワнтаイムID (SIGNALn, j) として生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバID、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワнтаイムIDを暗号化し、この暗号化データと上記第一のワнтаイムIDとを上記サーバに対して送信する送信手段と、上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワнтаイムID (SIGNAL' n, i) とし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワнтаイムID、上記第二の可変共有鍵、上記クライアントIDおよび上記サーバIDを引数とする一方向関数hの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方と、上記第二のワнтаイムIDとを上記サーバから受信する受信手段と、上記第二のワнтаイムIDを演算により求め、この演算結果と、上記サーバから受信した上記第二のワнтаイムIDとの照合により、上記サーバを識別し、上記サーバを識別

した場合に、上記サーバから受信した上記Diffie-Hellman公開値の他方と当該クライアントに予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数 $h$ の関数値を演算により求め、この演算結果と、上記サーバから受信した一方向関数 $h$ の関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、認証システムを、上記サーバと、上記クライアントとから構成したことを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムID (S I G N A L c<sub>1</sub>) として、この第一のワンタイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムID (S I G N A L s<sub>1</sub>) として求め、この第二のワンタイムIDと上記第二の乱数とを上記クライアントに対して送信する送信手段と、上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワンタイムID (S I G N A L c<sub>2</sub>) として、この第三のワンタイムIDを上記クライアントから受信する第二受信手段と、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、

所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成し、このワнтаイムIDを用いてサーバとの間で相互に認証を行うクライアントであって、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワнтаイムID (S I G N A L c<sub>1</sub>) として求め、この第一のワнтаイムIDと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムID (S I G N A L s<sub>1</sub>) として、この第二のワнтаイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワнтаイムIDとの比較により、上記サーバの正当性を判定する判定手段と、上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワнтаイムID (S I G N A L c<sub>2</sub>) として求め、この第三のワнтаイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして上記サーバと、上記クライアントとを備えてなることを特徴とするものである。

本発明はまた、請求項29に記載の発明は、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成し、このワнтаイムIDを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワнтаイムID (S I G N A L c<sub>1</sub>) として、この第一のワнтаイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、第二の乱数を生成するとともに

に、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムID (S I G N A L<sub>s1</sub>) として求め、この第二のワンタイムIDと上記第二の乱数を上記クライアントに対して送信する送信手段と、上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムID (S I G N A L<sub>c2</sub>) として、この第三のワンタイムIDを上記クライアントから受信する第二受信手段と、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いてサーバとの間で相互に認証を行うクライアントであって。第1の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムID (S I G N A L<sub>c1</sub>) として求め、この第一のワンタイムIDと上記第一の乱数を上記サーバに対して送信する第一送信手段と、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムID (S I G N A L<sub>s1</sub>) として、この第二のワンタイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムIDとの比較により、上記サーバの正当性を判定する判定手段と、上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムID (S I G N A L<sub>c2</sub>) として求め、この第三のワンタイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして上記サーバと、上記クライアントとを備え

てなることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムID (S I G N A L c i ) として、この第一のワンタイムIDを上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバIDを上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、上記第一のワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアントIDおよび上記サーバIDに基づいて、上記クライアントの正当性を判定する判定手段と、上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムID (S I G N A L s i ) として求め、上記クライアントID、上記サーバIDおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムIDとを上記クライアントに対して送信する送信手段と、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするものである。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いてサーバとの間で相互に認証を行うクライアントであって、第一

の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムID (S I G N A L<sub>ci</sub>) として求め、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムIDとを上記サーバに対して送信する送信手段と、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムID (S I G N A L<sub>si</sub>) として、この第二のワンタイムIDを上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアントIDおよび上記サーバIDを上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、上記第二のワンタイムIDを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムIDとの照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバIDおよび上記クライアントIDに基づいて、上記サーバの正当性を判定する判定手段と、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするものである。

本発明はまた、認証システムとして上記サーバと、上記クライアントとを備えてなることを特徴とするものである。

本発明はまた、上記認証システムにおいて、上記サーバおよび上記クライアントは、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後で、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようになっていることを特徴とするものである。

本発明に係る、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムID (S I G N A L<sub>n</sub>) を用いて、互いに通信を行なう第一装置と第二装置間における認証を行う認証方法によれば、可変共有鍵を引数と

する一方向関数の関数値を求め、この関数値からワンタイムIDを生成するようにしたため、例えば、可変共有鍵が第三者に漏れたとしても、所定の通信単位毎に可変共有鍵が変化することとなるので、漏れた可変共有鍵を用いて生成されたワンタイムID以外のワンタイムIDを予測することはできない。すなわち、盗聴が困難で安全性に優れたワンタイムIDを生成することが可能になり、ワンタイムIDの将来にわたる安全性（PFS）を実現することが可能になる。

本発明ではまた、可変共有鍵を生成し、且つ可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いて、第一装置と第二装置間における認証を行うようにしたため、例えば、可変共有鍵が第三者に漏れたとしても、所定の通信単位毎に可変共有鍵が変化するとともに、各通信毎に通信順序または回数に関する情報も変化することとなるので、漏れた可変共有鍵を用いて生成されたワンタイムID以外のワンタイムIDを予測することは事実上不可能となり、また漏れた可変共有鍵を用いて生成されたワンタイムIDの予測自体も非常に困難なものとなる。すなわち、盗聴が困難で安全性に優れたワンタイムIDを生成することが可能になり、ワンタイムIDの将来にわたる安全性（PFS）を実現することが可能になる。

本発明はまた、装置間またはアプリケーション間で所定の可変共有鍵を生成し、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、このワンタイムIDを用いて、第一装置と第二装置間における認証（相互認証）を行うようにしたため、例えば、共有鍵が第三者に漏れたとしても、乱数によって一方向関数の関数値が所定の通信単位毎に変化することとなるので、所定の通信単位内で生成される乱数がわからない限り、ワンタイムIDを予測することはできない。すなわち、盗聴が困難で安全性に優れたワンタイムIDを生成することが可能になり、ワンタイムIDの将来にわたる安全性（PFS）を実現することが可能になる。

本発明はまた、上述の種々のワンタイムIDの生成方法により生成されたワン



タイムIDを用いて、装置間（クライアント・サーバ間）における認証を行うようにしたので、第三者（攻撃者）が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムIDを識別情報として把握することができる。

したがって、DOS攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることができる。

本発明ではまた、第一装置の正当性を判定するのに用いる一方向関数 $F_c$ として、所定の共有鍵、Diffie-Hellman公開値の一方、第一装置に予め設定されたID、ワンタイムIDを引数とする疑似乱数関数を用いるとともに、第二装置の正当性を判定するのに用いる一方向関数 $F_s$ として、所定の共有鍵、Diffie-Hellman公開値の一方、Diffie-Hellman公開値の他方、第二装置に予め設定されたID、ワンタイムIDを引数とする疑似乱数関数を用いるようにしたので、従来の鍵交換・認証方式では3回必要であった通信回数を2回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

本発明はさらに、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成したり、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、装置間またはアプリケーション間で所定の可変共有鍵を生成したり、或いは、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成するなどの種々のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、装置間（クライアント・サーバ間）における認証を行うようにしたので、盗聴が困難で安全性に優れたワンタイムIDを生成することが可能となり、ワンタイムIDの将来にわたる安全性（PFS）を実現できるといった効果が得られる。

本発明はさらに、通信単位毎に変化する可変共有鍵を生成し、且つこの可変共

有鍵を引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成したり、可変共有鍵を生成し、且つ可変共有鍵と通信順序に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成し、装置間またはアプリケーション間で所定の可変共有鍵を生成したり、或いは、所定の通信単位内で乱数を生成し、且つこの乱数と前記共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワンタイムIDを生成するなどの種々のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、装置間（クライアント・サーバ間）の認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムIDを識別情報として把握できるといった効果が得られる。

したがって、DoS攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

かかる本発明の目的及び利点は添付図面を参照して説明される、以下の実施の形態によってより一層明らかなになるであろう。

#### 図面の簡単な説明

図1は、P-SIGMAと呼ばれる従来の認証方法を説明する図である。

図2は、本発明の実施の形態に係るクライアント・コンピュータとサーバ・コンピュータとの概略構成を示すブロック図である。

図3は、本発明の実施の形態に係る相互認証における概念プロセスを示すフローチャートである。

図4は、本発明の実施の形態に係る相互認証における詳細プロセスを示すイメージ図である。

図5は、本発明に係る認証システムの一実施形態を示す概略構成図である。

図6は、図1のサーバの概略構成を示すブロック図である。

図7は、図1のクライアントの概略構成を示すブロック図である。

図8は、本発明に係る認証方法の第1の実施形態を説明する図である。

図9は、本発明に係る認証方法の第2の実施形態を説明する図である。

図10は、本発明に係る認証方法の第3の実施形態を説明する図である。

図11は、本発明に係る認証方法の第4の実施形態を説明する図である。

図12は、本発明に係る認証方法の第5の実施形態を説明する図である。

図13は、本発明に係る認証方法の第6の実施形態を説明する図である。

図14は、OSPAと呼ばれる従来の認証方法を説明する図である。

図15は、本発明に係る認証方法の第7の実施形態を説明する図である。

図16は、図15の変形例を説明する図である。

## 発明を実施するための最良の形態

### (実施の形態1)

以下、図面を参照して本発明の実施の形態の一例を詳細に説明する。図2は本発明の第1の実施の形態に係るクライアント・コンピュータとサーバ・コンピュータとの概略構成、および本発明が適用可能なネットワーク・システムの概略構成を示すブロック図である。本実施の形態は、ネットワークにおいてサーバ・コンピュータとクライアント・コンピュータとの間で相互認証する場合に本発明を適用したものである。

図2において、ネットワークシステムは、CPUを少なくとも含む一または複数のクライアント・コンピュータ10、及びCPUを少なくとも含む一または複数のサーバ・コンピュータ40が、それぞれモデム、ルータ、TA（ターミナル・アダプタ：Terminal Adapter）等を介して、ネットワーク（例えば、インターネット）32に接続されて構成されている。これらのコンピュータは、ネットワーク32を介して、相互通信により情報授受が可能である。

なお、図2に示すように、クライアント・コンピュータ10及びサーバ・コンピュータ40の各々は1つのコンピュータとして説明するが、これらのクライアント・コンピュータ10、サーバ・コンピュータ40は複数台でもよい。

なお、クライアント・コンピュータ10が本発明の第1認証装置に相当するとき、サーバ・コンピュータ40が第2認証装置に相当し、サーバ・コンピュータ40が本発明の第1認証装置に相当するとき、クライアント・コンピュータ10

が第2認証装置に相当する。また、ネットワーク32は本発明の通信回線に相当する。

本実施の形態では、ネットワークとしてインターネットを適用した場合を説明する。この場合、少なくとも1つのコンピュータは、WWWサーバとして機能させることができ、また他のマシンはWWWクライアントとして機能させることもできる。

詳細には、各クライアント・コンピュータ10には、WWWブラウザがインストールされており、このWWWブラウザを起動することにより、ネットワーク32を介してサーバ・コンピュータ40に任意にアクセス可能となる。このとき、アクセス位置（アクセス先のサーバ・コンピュータ40の位置、及びサーバ・コンピュータ40内の情報の位置で構成されるデータ）は、URL（Uniform Resource Locator）で指定される。

サーバ・コンピュータ40は、クライアント・コンピュータ10からアクセス要求があった場合、URLで指定された位置にあるデータを、ネットワーク32を介して、アクセス元のクライアント・コンピュータ10へ送信する。このとき、データは、一般に、HTTPに従って転送される。

なお、クライアント・コンピュータ10の識別には、IP（Internet Protocol）アドレスが用いられる。また、クライアント・コンピュータ10を操作するユーザの識別には、ユーザ自身の入力や、予め定められているコード等のユーザIDを用いることができる。

上記コンピュータには、当該コンピュータで指示入力をするために、各々キーボード、マウス等の入力装置が設けられており、コンピュータによる処理結果等を表示するためにディスプレイが設けられている。なお、コンピュータは、汎用的かつ一般的なハードウェア構成であるため、詳細な説明を省略する。

クライアント・コンピュータ10は、システムパラメータ等を入力するための入力装置12を備えており、入力装置12は入力に応じた乱数Rを発生する乱数発生器14及びメモリ16に接続されている。乱数発生器14は、メモリ16及び乱数Rに基づく認証用データAを求める認証用データ演算器18に接続されて

いる。認証用データ演算器18は、ネットワーク32を介してサーバ・コンピュータ40と通信するためにネットワーク32に接続された通信インタフェース（以下、通信I/Fという）30に接続されている。

通信I/F30には、検証器20が接続されている。この検証器20はメモリ16及び認証用データ演算器18にも接続されている。また、検証器20は、サーバ・コンピュータ40との間で認証したときに、認証により相互関係が正当であると判定されたことを表示するOK装置22及び認証により相互関係が不当であると判定されたことを表示するNG装置24にも接続されている。

サーバ・コンピュータ40は、システムパラメータ等を入力するための入力装置42を備えており、入力装置42は入力に応じた乱数Qを発生する乱数発生器44及びメモリ46に接続されている。乱数発生器44は、メモリ46及び乱数Rに基づく認証用データBを求める認証用データ演算器48に接続されている。認証用データ演算器48は、ネットワーク32を介してクライアント・コンピュータ10と通信するために通信I/F60に接続されている。

通信I/F60には、検証器50が接続されている。この検証器50はメモリ46及び認証用データ演算器48にも接続されている。また、検証器50は、クライアント・コンピュータ10との間で認証したときに、認証により相互関係が正当であると判定されたことを表示するOK装置52及び認証により相互関係が不当であると判定されたことを表示するNG装置54にも接続されている。

#### 〔概念プロセス〕

次に、本実施の形態のネットワーク・システムにおける相互認証の概念プロセスを説明する。本実施の形態では、コンピュータ間の相互認証をデジタルデータの授受で実行する。図3には、相互認証の処理プロセスをフローチャートとして示した。

ステップ100では、クライアント・コンピュータ10及びサーバ・コンピュータ40は、予め定めた手順により、双方に共通な初期値（隠蔽鍵K。）を記憶する。

予め定めた手順とは、クライアント・コンピュータ10及びサーバ・コンピュ

ータ40の間の相互認証を実行するときの初期値を設定するものである。例えば、クライアント・コンピュータ10及びサーバ・コンピュータ40に共通なデータを初期値として保持させるため、クライアント・コンピュータ10及びサーバ・コンピュータ40の何れか一方、または第三者のコンピュータによって定められる初期値を、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方へ提供する。この提供は、初期値を電子メールなどの電子的にデータ送信することや、初期値を印刷した印刷物をクライアント・コンピュータ10及びサーバ・コンピュータ40の双方に送付してクライアント・コンピュータ10及びサーバ・コンピュータ40の各々で入力することによって達成される。

本実施の形態では、この初期値として、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方で共通な状態を維持するため、クライアント・コンピュータ10とサーバ・コンピュータ40との間でなされるデータ授受の履歴を初期値とし、後のクライアント・コンピュータ10とサーバ・コンピュータ40との間でなされるデータ授受毎に初期値を更新する。

すなわち、上記初期値は、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方に共通な値であればよく、上記のように任意の値を提供することで双方で保持してもよいが、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方で共通な状態を維持するため、任意のアルゴリズムによるクライアント・コンピュータ10及びサーバ・コンピュータ40の間のデータ授受の結果が好ましい。本実施の形態では、任意のアルゴリズムには、送信側と受信側との双方のデータを送信側及び受信側の双方で共通に保持する手順で可能であり、詳細を後述する相互認証の結果のデータを用いている。

なお、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方に記憶するデータの形式（例えばフォーマット）は、同一に限定するものではない。すなわち、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方に記憶するデータは、そのデータの最終的な値が同一であればよく、データそのものの同一性に限定されない。例えば、異なる形式で格納するようにしてもよい。このようにすれば、一方のデータが漏洩した場合であっても、他方のデ

ータは維持可能となる。

まず、ステップ110では、クライアント・コンピュータ10が、認証データを送付する。この認証データは、クライアント・コンピュータ10からサーバ・コンピュータ40に対して相互認証を要求する最初のデータであり、記憶されている初期値を隠蔽鍵として用い、クライアント・コンピュータ10内で生成されるデータを記憶すると共に隠蔽鍵による暗号化を行って、送付する。

次に、ステップ120では、サーバ・コンピュータ40において、クライアント・コンピュータ10から送付された認証データを受け取って、記憶されている初期値を隠蔽鍵として用い、この時点でサーバ・コンピュータ40内で生成されるデータを記憶すると共に隠蔽鍵による暗号化を行った認証データを送付する。なお、認証データには、クライアント・コンピュータ10から受け取った認証データに含まれる一部のデータを含ませる。

これにより、サーバ・コンピュータ40から送付する認証データがクライアント・コンピュータ10からの要求に対する応答であることを表すデータとして送付することができる。この認証データを送付した後は、受け取った認証データを解析すると共に、サーバ・コンピュータ40内で生成したデータの各々を用いて新規の隠蔽鍵を生成すると共に、新規の隠蔽鍵で、記憶されている隠蔽鍵を更新する。

次に、ステップ130では、クライアント・コンピュータ10において、サーバ・コンピュータ40から送付された認証データを受け取って、記憶されている初期値を隠蔽鍵として用い、この時点でクライアント・コンピュータ10内で生成されるデータを記憶すると共に隠蔽鍵による暗号化を行った認証データを送付する。なお、認証データには、サーバ・コンピュータ40から受け取った認証データに含まれる一部のデータを含ませる。

これにより、クライアント・コンピュータ10から送付する認証データがサーバ・コンピュータ40から送付されたものに対する応答であることを表すデータとして送付することができる。この認証データを送付した後は、受け取った認証データを解析すると共に、クライアント・コンピュータ10内で生成したデー

タの各々を用いて新規の隠蔽鍵を生成すると共に、新規の隠蔽鍵で、記憶されている隠蔽鍵を更新する。

従って、ステップ130のプロセスが終了した時点で、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方において、初期値（隠蔽鍵）が更新されて、共通の値（隠蔽鍵）として維持することができる。

次のステップ140では、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方のプロセスが予め定めた所定回数を完了したか否かを判断する。この判断基準回数は、少なくとも1回の回数が予め設定されており、本実施の形態では、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方に共通の回数の値が保持される。なお、判断基準回数は、クライアント・コンピュータ10及びサーバ・コンピュータ40の各々で異なる回数の値を保持してもよい。この場合には、クライアント・コンピュータ10及びサーバ・コンピュータ40の各々で認証の基準が異なることになるが、認証が正当であれば判断基準回数が少ないコンピュータ側で複数回のデータ授受が要求されることのみで達成できる。この回数を参照することで、クライアント・コンピュータ10では、ステップ140の更新処理、サーバ・コンピュータ40ではステップ120の更新処理が保持されている回数を終了するまで否定される。判断基準回数が1回に設定されている場合には、ステップ140で否定されることなく、そのままステップ150へ進む。

従って、ステップ140で肯定判断された時点で、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方において、共に値（隠蔽鍵）が更新されて、双方で共通の値（隠蔽鍵）が維持されることになる。すなわち、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方で保持する隠蔽鍵が情報授受毎に新規なものに更新され、常時最新の隠蔽鍵として維持することができる。

ステップ150では、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方において、認証処理が実行されて本プロセスを終了する。

上記認証処理は、記憶されている最新の隠蔽鍵を用いて、送付された認証デー



タが正当かデータであるか否かを判別することによってなされる。この認証処理は、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方において共通に実行できる。この認証処理が完了すると、クライアント・コンピュータ10及びサーバ・コンピュータ40の双方において相互認証が完了したことになる。

#### 〔詳細プロセス〕

次に、上記概念プロセスで述べた相互認証を詳細に説明する。

#### （隠蔽鍵を含むデータの構成）

本実施の形態では、隠蔽鍵は、情報授受毎に最新データに更新されるため、履歴データKとして機能する。以下の説明では、この履歴データKとして機能するものとして隠蔽鍵Kを同一表記とする。

上記概念プロセスで認証データとして用いる初期値を含む隠蔽鍵Kは、クライアント・コンピュータ10を特定するための暗証データC及び認証データRと、サーバ・コンピュータ40を特定するための暗証データS及び認証データQと、から構成される。なお、以下の説明では、隠蔽鍵K、暗証データC、認証データR、暗証データS及び認証データQに初期値「0」から増加する添え字を付し、更新状態を表すものとするが、これらを一般的に説明する場合には添え字を削除した記号のみを用いて説明する。

本実施の形態では、初期値として、詳細を後述するクライアント・コンピュータ10及びサーバ・コンピュータ40の双方でなされたデータ授受の結果を記憶するものとし、既に履歴データが内在するものとする。

隠蔽鍵Kは、暗証データC、認証データR、暗証データS及び認証データQの各々を用いた関数 $g(C, S, Q, R)$ の計算結果を用いる。関数 $g$ は、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。

また、クライアント・コンピュータ10側の初期値C、R。を生成するための最初の値は、暗証データC及び認証データRについてユーザが設定した値を用いてもよく、自動的に生成してもよい。認証データRは情報授受毎に内容が無規則で変動することが好ましいので、本実施の形態では、認証データRとして乱数

発生器 14 で発生された乱数を用いている。しかし、本発明は、認証データ R に乱数を用いることに限定されるものではない。例えば、現在年月日、日時、時刻などの時間データ、コンピュータ内に格納された任意ファイル容量やタイムスタンプ、及び情報授受のときの容量などを用いることができる。

同様に、サーバ・コンピュータ 40 側の初期値  $S_0$ 、 $Q_0$  を生成するための最初の値は、暗証データ S 及び認証データ Q についてサーバ・コンピュータ 40 を管理するオペレータが設定した値を用いてもよく、自動的に生成してもよい。上記と同様に認証データ Q は情報授受毎に内容が無規則で変動することが好ましいので、本実施の形態では、認証データ Q として乱数発生器 44 で発生された乱数を用いている。しかし、本発明は、認証データ Q に乱数を用いることに限定されるものではない。例えば、現在年月日、日時、時刻などの時間データ、コンピュータ内に格納された任意ファイル容量やタイムスタンプ、及び情報授受のときの容量などを用いることができる。

また、クライアント・コンピュータ 10 側の認証データ R、及びサーバ・コンピュータ 40 側の認証データ Q を他方へ送信するが、その送信データについて第三者による特定を困難にするため、秘匿する必要がある。そこで、本実施の形態では、クライアント・コンピュータ 10 からサーバ・コンピュータ 40 へ送信する認証データ R、及びサーバ・コンピュータ 40 からクライアント・コンピュータ 10 へ送信する認証データ Q を隠蔽鍵 K で隠蔽する。

すなわち、クライアント・コンピュータ 10 からサーバ・コンピュータ 40 へ送信する場合、予め定めた関数  $v(R, K)$  により認証データ A を生成して送信する。関数  $v$  は、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。同様に、サーバ・コンピュータ 40 からクライアント・コンピュータ 10 へ送信する場合も、予め定めた関数  $w(Q, K)$  により認証データ B を生成して送信する。関数  $w$  は、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。次に、関数  $v$ 、 $w$  の一例を示す。

$$A_m = v(R, K) = R_m + K_{m-1}$$

$$B_m = w(Q, K) = Q_m + K_{m-1}$$

ただし、 $m \geq 1$  の自然数である。

また、クライアント・コンピュータ10側の暗証データC、及びサーバ・コンピュータ40側の暗証データSを他方へ送信するが、以下に説明するように、暗証データは情報授受の度に変更している。すなわち、クライアント・コンピュータ10からサーバ・コンピュータ40へ送信する暗証データCは、その送信するときに予め定めた関数 $y(S, R)$ により新規の暗証データCを生成して送信する。関数 $y$ は、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。同様に、サーバ・コンピュータ40からクライアント・コンピュータ10へ送信する場合も、予め定めた関数 $z(C, Q)$ により暗証データSを生成して送信する。関数 $z$ は、単純和や係数付加の多項式、乗算、積和そしてハッシュ関数が一例としてある。次に、関数 $y$ 、 $z$ の一例を示す。

$$C_m = y(S, R) = S_{m-1} + R_{m-1}$$

$$B_m = w(C, Q) = C_{m-1} + Q_{m-1}$$

ただし、 $m \geq 1$  の自然数である。

なお、暗証データの送信では、第三者による特定を困掛こするため、秘匿してもよい。例えば、クライアント・コンピュータ10からサーバ・コンピュータ40へ送信する暗証データC、及びサーバ・コンピュータ40からクライアント・コンピュータ10へ送信する暗証データSを隠蔽鍵Kで隠蔽するようにしてもよい。すなわち、隠蔽鍵Kをパラメータとして追加した関数にしてもよい。

(詳細プロセス)

図4は本発明の第1の実施の形態に係る相互認証における詳細プロセスを示すイメージ図である。以下、図4を参照して本実施の形態の詳細プロセスを説明する。

ステップP0：クライアント・コンピュータ10及びサーバ・コンピュータ40の各々に、初期値の隠蔽鍵 $K_0$ を格納する。このプロセスは、図3のステップ100と、図4のプロセスPc0及びPs0に相当する。

ステップP1：クライアント・コンピュータ10では、乱数Rを生成し、暗証データC及び認証データAを計算し、サーバ・コンピュータ40に送信する。こ

のプロセスは、図3のステップ110と、図4のプロセスPc1に相当する。

すなわち、クライアント・コンピュータ10においては、乱数発生器14において乱数 $R_1$ を生成する。生成された乱数 $R_1$ 、メモリ16に記憶されている隠蔽鍵 $K_0$ 及び隠蔽鍵 $K_0$ を構成する $C_0$ 、 $S_0$ 、 $Q_0$ 、 $R_0$ は認証用データ演算器18へ入力される。そして、認証用データ演算器18は、その乱数 $R_1$ 及びメモリ16に記憶されている隠蔽鍵 $K_0$ 及び隠蔽鍵 $K_0$ を構成する暗証データ $S_0$ 、認証データ $R_0$ を用いて上記関数 $y$ 、 $v$ により新規の暗証データ $C_1$ 、及び新規の認証データ $A_1$ を求める。この求めた新規の暗証データ $C_1$ 、及び認証データ $A_1$ はメモリ16に記憶すると共に、通信I/F30に出力され、ネットワーク32を介してサーバ・コンピュータ40へ送信される。この送信データは、図4のデータDc1に相当する。

ステップP2：サーバ・コンピュータ40は、クライアント・コンピュータ10から認証データ $A$ 及び暗証データ $C$ を受信すると共に、乱数 $Q$ を生成し暗証データ $S$ 、認証データ $Q$ を計算しクライアント・コンピュータ10に送信する。これと共に、記憶されている隠蔽鍵 $K_0$ を新規の隠蔽鍵 $K_1$ に更新する。このプロセスは、図3のステップ120と、図4のプロセスPs1に相当する。

すなわち、サーバ・コンピュータ40では、通信I/F60を介して検証器50にクライアント・コンピュータ10からの暗証データ $C_1$ 及び認証データ $A_1$ が入力される。このとき、サーバ・コンピュータ40では、乱数発生器44において乱数 $Q_1$ を生成する。生成された乱数 $Q_1$ 、メモリ46に記憶されている隠蔽鍵 $K_0$ 及び隠蔽鍵 $K_0$ を構成する $C_0$ 、 $S_0$ 、 $Q_0$ 、 $R_0$ は認証用データ演算器48へ入力される。また、検証器50は、クライアント・コンピュータ10からの暗証データ $C_1$ 及び認証データ $A_1$ を認証用データ演算器48へ出力する。

認証用データ演算器48は、その乱数 $Q_1$ 、受信した暗証データ $C_1$ 及び記憶されている隠蔽鍵 $K_0$ 及び隠蔽鍵 $K_0$ を構成する認証データ $Q_0$ を用いて上記関数 $z$ 、 $w$ により新規の暗証データ $S_1$ 、及び新規の認証データ $B_1$ を求める。この求めた新規の暗証データ $S_1$ 、及び認証データ $B_1$ は通信I/F60に出力され、ネットワーク32を介してクライアント・コンピュータ10へ送信される。

この送信データは、図4のデータDs1に相当する。

このとき、サーバ・コンピュータ40では、初期値としての隠蔽鍵 $K_0$ を構成する各データについて新規のデータが入手できている。すなわち、暗証データCについてはクライアント・コンピュータ10から受信した暗証データ $C_1$ 、暗証データSについては認証用データ演算器48で計算した暗証データ $S_1$ 、認証データQについては乱数発生器44で発生した乱数 $Q_1$ 、認証データRについてはクライアント・コンピュータ10から受信した認証データAから逆算すなわち隠蔽鍵 $K_0$ を減算することで得られる乱数 $R_1$ である。

そこで、これらの暗証データ $C_1$ 、暗証データ $S_1$ 、認証データ $Q_1$ 、認証データ $R_1$ を、新規のデータとして更新すると共に、新規の隠蔽鍵 $K_1$ として更新する。これによって、サーバ・コンピュータ40では、隠蔽鍵 $K$ の履歴として最新のデータに自動的に更新することができる。

ステップP3：クライアント・コンピュータ10は、サーバ・コンピュータ40から認証データB及び暗証データSを受信すると共に、乱数Rを生成し暗証データ $C_2$ 、認証データ $A_2$ を計算しサーバ・コンピュータ40に送信する。これと共に、記憶されている隠蔽鍵 $K_0$ を新規の隠蔽鍵 $K_1$ に更新する。このプロセスは、図3のステップ130と、図4のプロセスPc2に相当する。

すなわち、クライアント・コンピュータ10では、通信I/F30を介して検証器20にサーバ・コンピュータ40からの暗証データ $S_1$ 及び認証データ $B_1$ が入力される。このとき、クライアント・コンピュータ10では、乱数発生器14において乱数 $R_2$ を生成する。生成された乱数 $Q_2$ 、メモリ46に記憶されている隠蔽鍵 $K_0$ 及び隠蔽鍵 $K_0$ を構成する $C_0$ 、 $S_0$ 、 $Q_0$ 、 $R_0$ は認証用データ演算器18へ入力される。また、検証器20は、サーバ・コンピュータ40からの暗証データ $S_1$ 及び認証データ $B_1$ を認証用データ演算器18へ出力する。

このとき、クライアント・コンピュータ10では、初期値としてメモリ16に記憶されている隠蔽鍵 $K_0$ を構成する各データについて新規のデータ（新規の隠蔽鍵 $K_1$ を構成するデータ）が入手できている。すなわち、暗証データCについてはサーバ・コンピュータ40から受信した暗証データ $S_1$ から逆算すなわち隠

蔽鍵 $K_0$ を構成したメモリ16に記憶されている認証データ $Q_0$ を減算することで得られる暗証データ $C_1$ 、またはメモリ16に記憶されている前回送信した暗証データ $C_1$ が対応する。暗証データ $S$ についてはサーバ・コンピュータ40から受信した暗証データ $S_1$ 、認証データ $Q$ についてはサーバ・コンピュータ40から受信した認証データ $B_1$ から逆算すなわち隠蔽鍵 $K_0$ を減算することで得られる認証データ $Q_1$ 、認証データ $R$ については前回生成した乱数 $R_1$ である。

そこで、これらの暗証データ $C_1$ 、暗証データ $S_1$ 、認証データ $Q_1$ 、認証データ $R_1$ を、新規のデータとして更新すると共に、新規の隠蔽鍵 $K_1$ として更新する。これによって、クライアント・コンピュータ10では、サーバ・コンピュータ40と同一の隠蔽鍵 $K$ を、最新のデータに自動的に更新することができる。また、認証用データ演算器18は、生成した乱数 $R_2$ 、更新した履歴データ $K_1$ の認証データ $R_1$ 、受信した暗証データ $S_1$ 及び新規の隠蔽鍵 $K_1$ を用いて上記関数 $y$ 、 $v$ により新規の暗証データ $C_2$ 、及び新規の認証データ $A_2$ を求める。この求めた新規の暗証データ $C_2$ 、及び認証データ $A_2$ はメモリ16に記憶すると共に、通信I/F30に出力され、ネットワーク32を介してサーバ・コンピュータ40へ送信される。この送信データは、図4のデータDc2に相当する。

ステップP4：上記ステップP2及びP3のプロセスを所定回数 $m$ だけ実行する。なお、本実施の形態では、所定回数 $m$ は、少なくとも1回のデータ授受を含む。このため、繰り返しを行わない回数( $m=1$ )を含むものである。すなわち、クライアント・コンピュータ10とサーバ・コンピュータ40との間でなされるデータ授受のときには、既に双方でなされたデータ授受の履歴データが利用されるため、1回のデータ授受であっても、その授受のときにはクライアント・コンピュータ10とサーバ・コンピュータ40との間の履歴を含んでデータ授受がなされるため、単なるデータ授受ではなく、履歴データの授受となるので有効である。上記ステップP2及びP3のプロセスを複数回数だけ繰り返すことは、データの正当性の判断精度向上に有効である。

すなわち、上記処理を繰り返すプロセスは、繰り返す回数すなわち実行回数を複数回予め定めておくことで、隠蔽鍵 $K$ の値が更新されることで変動し、その変

動を第三者が把握することを抑制することが可能になる。このように複数回とすることで、クライアント・コンピュータ10及びサーバ・コンピュータ40で共通に保持する隠蔽鍵Kは複数回数だけ今までの履歴に従って最新の状態に更新されるので、隠蔽鍵Kを導出することが困難になる。

ステップP2及びP3のプロセスを所定回数 $m$ だけ実行した結果、クライアント・コンピュータ10及びサーバ・コンピュータ40の各々には、隠蔽鍵 $K_m$ 及び隠蔽鍵 $K_m$ を構成する $C_m$ ,  $S_m$ ,  $Q_m$ ,  $R_m$ の値が保持される。なお、 $m=1$ のときは、1回のデータ授受の値が保持される。

なお、処理を繰り返す実行プロセスは、図3のステップ140の判断によるプロセス実行と、図4のプロセスPc2からPsm及びPcmのプロセスについて、プロセスPc1からPs1及びPc2のプロセスを繰り返したことに相当する。

ステップP5：上記プロセスが終了した後に、クライアント・コンピュータ10及びサーバ・コンピュータ40の各々では、受信したデータの正当性が成立するか否かを検査し、成立すれば相互認証が成功したものととして両者の関係を許諾し、非成立のときは相互認証が不成功であるとして両者の関係を拒否する。このプロセスは、図3のステップ150と、図4のプロセスPsm+1及びPcm+1に相当する。

1回の実行の後に認証する場合には、クライアント・コンピュータ10から1回目のデータ送信がなされるが、そのときクライアント・コンピュータ10は、クライアント・コンピュータ10とサーバ・コンピュータ40との履歴を含む初期値として記憶された隠蔽鍵 $K_0$ により生成される、認証データ $A_1$ 、及び暗証データ $C_1$ をサーバ・コンピュータ40へ送信する。このプロセスは、図4のプロセスPc1の後にデータDc1を送信することに相当する。

サーバ・コンピュータ40では、通信I/F60を介して検証器50にクライアント・コンピュータ10からの暗証データ $C_1$ 及び認証データ $A_1$ が入力され、暗証データ $C_1$ について検証器50において正当性を検証する。受信した暗証データ $C_1$ は、前回の履歴のデータに基づいて生成されているため、サーバ・コンピュータ40では、最新の状態に更新記憶されている隠蔽鍵 $K$ 。（ここでは初期

値)を構成する暗証データ $S$ 。及び認証データ $R$ 。を用いて上述の関数 $y$ の計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、OK装置52で正当性があることを報知した後に処理を継続し、否認されたときにはNG装置54で不当であることを報知した後に処理を終了する。

正当性が認められて処理が継続されたときには、上記ステップP2と同様にして、乱数発生器44において乱数 $Q_1$ を生成し、認証用データ演算器48において暗証データ $S_1$ 、認証データ $B_1$ を生成して、クライアント・コンピュータ10へ送信すると共に、隠蔽鍵を最新の隠蔽鍵 $K_1$ に更新する。

この認証プロセスは、図4のプロセス $P_{s_{m+1}}$ の処理に相当する。この場合、繰り返して実行していないので、 $m=0$ で処理したことに相当する。すなわち、クライアント・コンピュータ10からサーバ・コンピュータ40へデータを送信する毎に、サーバ・コンピュータ40側でクライアント・コンピュータ10から受け取った履歴を含むデータを用いて認証を行うことができる。

一方、クライアント・コンピュータ10では、通信I/F30を介して検証器20にサーバ・コンピュータ40からの暗証データ $S_1$ 及び認証データ $B_1$ が入力される。クライアント・コンピュータ10では、暗証データ $S_1$ について検証器20において正当性を検証する。受信した暗証データ $S_1$ は、暗証データ $C$ と同様に前回の履歴のデータに基づいてサーバ・コンピュータ40において生成されているため、クライアント・コンピュータ10では、最新の状態に更新記憶されている隠蔽鍵 $K$ 。(ここでは初期値)を構成する暗証データ $C$ 。及び認証データ $Q$ 。を用いて上述の関数 $z$ の計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、OK装置22で正当性があることを報知した後に処理を継続し、否認されたときにはNG装置24で不当であることを報知した後に処理を終了する。

正当性が認められて処理が継続されたときには、クライアント・コンピュータ10とサーバ・コンピュータ40との間で実行すべき処理へと移行する。なお、



クライアント・コンピュータ10では、サーバ・コンピュータ40との履歴データKの同一性を維持するため、上記ステップP3と同様にして、隠蔽鍵を最新の隠蔽鍵 $K_1$ に更新する。

この認証プロセスは、図4のプロセス $P_{c_{m+1}}$ の処理に相当する。この場合、繰り返して実行していないので、 $m=0$ で処理したことに相当する。すなわち、サーバ・コンピュータ40からクライアント・コンピュータ10へデータを送信する毎に、クライアント・コンピュータ10側でサーバ・コンピュータ40から受け取った履歴を含むデータを用いて認証を行うことができる。

なお、クライアント・コンピュータ10からサーバ・コンピュータ40へデータを送信する毎に、またはサーバ・コンピュータ40からクライアント・コンピュータ10へデータを送信する毎に、受け取り側で認証を行うことを含めた処理をセッションとして、この認証を含めたセッションを複数回実行してもよい。

次に、複数回の実行を繰り返した後に認証する場合を説明する。この場合には、クライアント・コンピュータ10から $m$ 回目のデータ送信がなされ、クライアント・コンピュータ10は、 $m$ 回の繰り返しによって更新された隠蔽鍵 $K_m$ 、により、サーバ・コンピュータ40へ認証データ $A_{m+1}$ 、及び暗証データ $C_{m+1}$ を送信する。このプロセスは、図4のプロセス $P_{c_m}$ の後にデータ $D_{c_{m+1}}$ を送信することに相当する。

まず、サーバ・コンピュータ40では、通信I/F60を介して検証器50にクライアント・コンピュータ10からの暗証データ $C_{m+1}$ 及び認証データ $A_{m+1}$ が入力される。サーバ・コンピュータ40では、暗証データ $C_{m+1}$ について検証器50において正当性を検証する。受信した暗証データ $C_{m+1}$ は、前回の履歴のデータに基づいて生成されているため、サーバ・コンピュータ40では、最新の状態に更新記憶されている隠蔽鍵 $K_m$ を構成する暗証データ $S_m$ 及び認証データ $R_m$ を用いて上述の関数 $y$ の計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、OK装置52で正当性があることを報知した後に処理を継続し、否認されたときにはNG装置54で不当であることを報知した後に処理を

終了する。

正当性が認められて処理が継続されたときには、上記ステップ P 2 と同様にして、乱数発生器 4 4 において乱数  $Q_{m+1}$  を生成し、認証用データ演算器 4 8 において暗証データ  $S_{m+1}$ 、認証データ  $B_{m+1}$  を生成して、クライアント・コンピュータ 1 0 へ送信すると共に、隠蔽鍵を最新の隠蔽鍵  $K_{m+1}$  に更新する。この認証プロセスは、図 4 のプロセス  $P_{S_{m+1}}$  の処理に相当する。

一方、クライアント・コンピュータ 1 0 では、通信 I/F 3 0 を介して検証器 2 0 にサーバ・コンピュータ 4 0 からの暗証データ  $S_{m+1}$  及び認証データ  $B_{m+1}$  が入力される。クライアント・コンピュータ 1 0 では、暗証データ  $S_{m+1}$  について検証器 2 0 において正当性を検証する。受信した暗証データ  $S_{m+1}$  は、暗証データ C と同様に前回の履歴のデータに基づいてサーバ・コンピュータ 4 0 において生成されているため、クライアント・コンピュータ 1 0 では、最新の状態に更新記憶されている隠蔽鍵  $K_m$  を構成する暗証データ  $C_m$  及び認証データ  $Q_m$  を用いて上述の関数  $z$  の計算結果と、受信したデータが一致するか否かを判別し、一致する場合は正当性を認め不一致の場合は正当性を否定する。正当性が認められたときには、OK 装置 2 2 で正当性があることを報知した後に処理を継続する一方、否認されたときには NG 装置 2 4 で不当であることを報知した後に処理を終了する。

正当性が認められて処理が継続されたときには、クライアント・コンピュータ 1 0 とサーバ・コンピュータ 4 0 との間で実行すべき処理へと移行する。なお、クライアント・コンピュータ 1 0 では、サーバ・コンピュータ 4 0 との履歴データ K の同一性を維持するため、上記ステップ P 3 と同様にして、隠蔽鍵を最新の隠蔽鍵  $K_{m+1}$  に更新する。この認証プロセスは、図 4 のプロセス  $P_{C_{m+1}}$  の処理に相当する。

このように、本実施の形態では、クライアント・コンピュータ 1 0 とサーバ・コンピュータ 4 0 との間の相互認証をするときに、双方で共通の隠蔽鍵 K を有し、その隠蔽鍵 K を情報授受毎に更新している。このため、情報授受のときのデータを解析しても、認証用のデータを特定することが困難であり、秘匿性を向上する

ことができ、確実に相互認証が可能となる。

上記では、クライアント・コンピュータ10とサーバ・コンピュータ40との間を例にして説明したが、インターネット等の非同期ネットワークにおいては、クライアント・コンピュータ10に対してサーバ・コンピュータ40では認証が必要である。この場合には、クライアント・コンピュータ10のユーザID毎に処理を分離するようにしてもよい。

上記プロセスは、クライアント・コンピュータ10及びサーバ・コンピュータ40の処理プログラムとして記録媒体としてのフレキシブルディスクに実行可能な形式で格納することができる。この場合、各装置に挿抜可能なフレキシブルディスクユニット(FDU)を接続して、フレキシブルディスクからFDUを介して記録された処理プログラムを実行すればよい。また、処理プログラムをコンピュータ内のRAMや他の記憶領域(例えばハードディスク装置)にアクセス可能に格納にして(インストール)して実行するようにしてもよい。また、予めROMに記憶してもよい。また、記録媒体としては、CD-ROM、MD、MO、DVD等のディスクやDAT等の磁気テープがあり、これらを用いるときには、対応する装置としてCD-ROM装置、MD装置、MO装置、DVD装置、DAT装置等を用いればよい。

以上説明したように本発明の第1の実施の形態によれば、第1認証装置及び第2認証装置の間で相互認証するときに、第1認証装置及び第2認証装置の各々に共通に履歴データを記憶すると共に、履歴データを更新するので、安全かつ簡便に相互認証することができ、例えば、クライアント・コンピュータとサーバ・コンピュータとの間で授受される情報から、クライアント・コンピュータの鍵が漏洩することがなく、確実に認証が行える、という効果がある。

#### (実施の形態2)

図5は、本発明に係る認証システムの第2の実施の形態を示す概略構成図である。この認証システムは、公衆回線網やインターネット等のネットワーク40を介して相互に接続されたサーバ(第二装置)10とクライアント(第一装置)20とにより概略構成されている。この実施の形態では、種々のサービスを提供す

る複数のサーバA、B、C、…がサーバ10に接続され、当該サーバ10が、サーバA、B、C、…へのアクセスの可否を決定する認証サーバとして機能するようになっている。

サーバ10は、図6に示すように、CPU11、RAM12、記憶装置13、入力装置14、表示装置15および通信装置16等により構成され、各部はバス17により接続されている。

CPU (Central Processing Unit) 11は、記憶装置13の記憶領域に格納されている各種処理プログラム、入力装置14や通信装置16から入力される各種指示、あるいは指示に対応する各種データ等をRAM12に格納し、それら入力指示および各種データに応じてRAM12に格納した各種処理プログラムに従って各種処理を実行し、その処理結果をRAM12に一時的に記憶するとともに、表示装置15等に出力する。

このCPU11は、当該サーバ10における受信手段および判定手段を構成しており、クライアントIDを引数とする一方向関数（一方向関数 $F_c$ ）の関数値であるHASH $c$ 、ワンタイムID (SIGNAL)、DH公開値 $g^*$  (Diffie-Hellman公開値の一方) をクライアント20から受信した場合（すなわち、クライアント20からアクセスの要求を受けた場合）に、クライアント20から受信した受信データと記憶装置13に記憶されている記憶データを用いてワンタイムIDおよびHASH $c$ を演算により求め、この演算結果と、クライアント20から受信したワンタイムIDおよびHASH $c$ との比較により、クライアント20の正当性を判定する処理を実行する。

また、CPU11は、当該サーバ10における送信手段を構成しており、クライアント20が正当であると判定される場合に、上記受信データおよび上記記憶データを用いて、サーバIDを引数とする一方向関数（一方向関数 $F_s$ ）の関数値であるHASH $s$ を演算により求め、このHASH $s$ と、記憶装置13に記憶されているDH公開値 $g^*$  (Diffie-Hellman公開値の他方) とをクライアント20に対して送信する処理を実行する。

なお、上記ワンタイムID (SIGNAL) は、サーバ・クライアント間にお

ける認証において一回限り使用可能な識別情報であり、このワンタイムIDを生成する場合には、所定の通信単位毎に変化する暗号化鍵K（可変共有鍵）を記憶装置13から読み込んで、この暗号化鍵Kを引数とするハッシュ関数（一方向関数）の関数値を求め、この関数値から上記ワンタイムIDを生成する。

RAM（Random Access Memory）12は、クライアント20等との間で送受信されるデータなど、認証に関する各種データを一時記憶する記憶領域や、CPU11の作業領域などを備えている。

記憶装置13は、プログラムやデータ等が記憶される記憶媒体13aを有し、この記憶媒体13aは磁氣的、光学的記録媒体、若しくは半導体メモリで構成されている。この記憶媒体13aは記憶装置13に固定的に設けたもの、若しくは着脱自在に装着するものであり、CPU11により実行される各種処理プログラムや制御データ等を記憶する記憶領域、認証に関する各種データ（例えば、クライアント20やID発行管理サーバ30（後述）から取得したデータ、認証の処理過程で生成されたデータなど）を格納する記憶領域などを備えている。なお、この記憶媒体13aに記憶するプログラムやデータなどは、その一部若しくは全部を他のサーバ等からネットワーク40を介して受信して記憶する構成とすることも可能である。この記憶媒体13aには、サーバID、DH公開値 $g^r$ 、クライアント20との間で共有化された乱数Rなどが、認証処理を開始する前段階で予め格納された状態となっている。

入力装置14は、キーボードやポインティングデバイス等により構成され、入力指示信号をCPU11に対して出力する。

表示装置15は、CRT（Cathode Ray Tube）やLCD（Liquid Crystal Display）等により構成され、CPU11から入力される表示データを表示する。通信装置16は、モデムやルータ、ブリッジ等により構成され、ネットワーク40を介してクライアント20等より受信したデータをCPU11に出力するとともに、CPU11より受信したデータをネットワーク40を介してクライアント20等に対して出力する。

一方、クライアント20は、図7に示すように、CPU21、RAM22、記

憶装置 23、入力装置 24、表示装置 25 および通信装置 26 等により構成され、各部はバス 27 により接続されている。具体的に、クライアント 20 としては、例えば、パーソナルコンピュータや、PDA (Personal Digital Assistance) 等の携帯情報端末、インターネット接続サービスを利用可能な携帯電話などが挙げられる。なお、このクライアント 20 の各構成要素は、前述したサーバ 10 の各構成要素とほぼ同様であるので、相違点のみを以下に説明する。

すなわち、クライアント 20 の CPU 21 は、当該クライアント 20 における送信手段を構成しており、入力装置 24 からの指示入力等に基づいて、ワンタイム ID (SIGNAL) を生成するとともに、クライアント ID を引数とする一方向関数 (一方向関数  $F_c$ ) の関数値である HASH  $c$  を求め、これらワンタイム ID および HASH  $c$  と、記憶装置 23 に予め記憶された DH 公開値  $g^x$  (Diffie-Hellman 公開値の一方) とをサーバ 10 に対して送信する処理を実行する。

また、CPU 21 は、当該クライアント 20 における受信手段および判定手段を構成しており、サーバ ID を引数とする一方向関数 (一方向関数  $F_s$ ) の関数値である HASH  $s$  と、DH 公開値  $g^y$  (Diffie-Hellman 公開値の他方) とをサーバ 10 から受信した場合 (すなわち、サーバ 10 によってクライアント 20 が正当であると判定された場合) に、サーバ 10 から受信した受信データと記憶装置 23 に記憶されている記憶データを用いて HASH  $s$  を演算により求め、この演算結果と、サーバ 10 から受信した HASH  $s$  との比較により、サーバ 10 の正当性を判定する処理を実行する。

記憶装置 23 は、プログラムやデータ等が記憶される記憶媒体 23a を有し、この記憶媒体 23a は、上記 CPU 21 により実行される各種処理プログラムや制御データ等を記憶する記憶領域、認証に関する各種データ (例えば、サーバ 10 や ID 発行管理サーバ 30 (後述) から取得したデータ、認証の処理過程で生成されたデータなど) を格納する記憶領域などを備えている。この記憶媒体 23a には、クライアント ID、DH 公開値  $g^x$ 、サーバ 10 との間で共有化された乱数  $R$  などが、認証処理を開始する前段階で予め格納された状態となっている。

ID 発行管理サーバ 30 は、クライアント・サーバ間で共有化される秘密情報

(例えば、ワンタイムIDの初期値を生成するのに用いられる乱数Rなど)や、クライアントID、サーバIDなどを発行・管理するためのサーバである。このID発行管理サーバ30は、クライアント20を利用するユーザのID(例えば、クレジットNo、住基ネットID、社員No、学生No、特定会員Noなど)に上記秘密情報やパスワードなどを対応付けた状態で格納するデータベースを有している。また、ID発行管理サーバ30は、一定の周期で上記データベース内の秘密情報を更新し、この更新した秘密情報をオンライン(例えば、電子メールなど)またはオフライン(例えば、郵送など)で、クライアント20とサーバ10の双方に配布するようになっている。なお、上記秘密情報の発行は、クライアント20またはサーバ10からの発行依頼に基づくものであってもよい。

次に、上記構成からなる認証システムによって行われる認証方法の第2の実施の形態について、図8に基づいて説明する。この方法は、RFC2409において規定されたIKEの方式に、本発明に係るワンタイムID(SIGNAL)を適用したものである。

まず、ステップS1では、IKEによるSA生成に際してイニシエータとなるクライアント20が、ワンタイムID(SIGNAL)を生成するとともに、HASHcを演算により求め、これらワンタイムIDおよびHASHcと、記憶装置23に記憶されたDH公開値 $g^x$ とをSAの提案とともに、レスポндаとなるサーバ10に対して送信する処理を実行する。

ここで、ワンタイムIDであるSIGNALは、例えば、ハッシュ関数を用いて、次のように生成される。

$$\text{SIGNAL}_1 = R$$

$$\text{SIGNAL}_2 = \text{hash}(K_1)$$

$$\text{SIGNAL}_3 = \text{hash}(K_2)$$

...

$$\text{SIGNAL}_n = \text{hash}(K_{n-1})$$

..... (式2)

上記S I G N A Lの定義式において、h a s hはハッシュ関数、RはI D発行管理サーバ30からサーバ10とクライアント20の双方に発行されて両者間で共有化された乱数、 $K_i$ はi番目のセッションで生成されたサーバ・クライアント共有の暗号化鍵（可変共有鍵）である。なお、上記セッションは、S Aを確立してから当該S Aが無効になるまでの通信単位を示している。

すなわち、上記S I G N A Lの定義式によれば、前回のセッションで生成された上記暗号化鍵Kを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのS I G N A Lとして用いるようにしている。また、最初のセッションでは、サーバ・クライアント間で予め共有化された乱数RをS I G N A Lの初期値として用いるようにしている。また、上記暗号化鍵 $K_i$ は、例えば、次式（3）から求められる。

$$K_i = \text{prf}(\text{共有鍵}, g^{xy}, \text{S I G N A L}_i) \quad \dots\dots\dots \text{(式3)}$$

この式（3）において、 $g^{xy}$ はDH共通鍵であり、共有鍵は、サーバ・クライアント間の任意の共有鍵である。

一方、H A S H cは、次式（4）に示すように、共有鍵、DH公開値 $g^x$ 、I D c（クライアントI D）およびS I G N A Lを引数とする疑似乱数関数（鍵付きハッシュ関数）の関数値として求められる。

$$\text{H A S H c} = \text{prf}(\text{共有鍵}, g^x, \text{I D c}, \text{S I G N A L}) \quad \dots\dots\dots \text{(式4)}$$

次いで、ステップS2では、サーバ10が、S I G N A LとH A S H cを演算により求め、これら演算結果と、クライアント20から受信したS I G N A LおよびH A S H cとの比較により、クライアント20の正当性を判定する処理を実行する。

上記判定の結果、受信データと演算結果とが一致して、クライアント20が正当であると判定される場合には、H A S H sを演算により求め、このH A S H sと、記憶装置13に記憶されているDH公開値 $g^y$ とを、受諾したS Aとともにクライアント20に対して送信する処理を実行する（ステップS3）。一方、受



信データと演算結果とが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

ここで、HASHsは、次式(5)に示すように、共有鍵、DH公開値 $g^x$ 、 $g^y$ 、IDs(サーバID)およびSIGNALを引数とする疑似乱数関数(鍵付きハッシュ関数)の関数値として求められる。

$$\text{HASHs} = \text{prf}(\text{共有鍵}, g^x, g^y, \text{IDs}, \text{SIGNAL})$$

..... (式5)

また、このステップS3においては、記憶装置13に記憶されているDH公開値 $g^y$ と、クライアント20から受信したDH公開値 $g^x$ とからDH共通鍵 $g^{xy}$ を生成して、DH共通鍵 $g^{xy}$ を記憶装置13に格納する処理も併せて行う。

次いで、ステップS4では、クライアント20が、HASHsを演算により求め、この演算結果と、サーバ10から受信したHASHsとの比較により、サーバ10の正当性を判定する処理を実行する。

上記判定の結果、受信データと演算結果とが一致して、サーバ10が正当であると判定される場合には、記憶装置23に記憶されているDH公開値 $g^x$ と、サーバ10から受信したDH公開値 $g^y$ とからDH共通鍵 $g^{xy}$ を生成して記憶装置23に格納した後、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、サーバ10が正当でないと判定される場合には、サーバ10へのアクセスを中止して、当該認証処理を終了する。

以上のように、この第2の実施の形態によれば、セッション毎に変化する暗号化鍵K(可変共有鍵)を引数とするハッシュ関数の関数値をワンタイムID(SIGNAL)として用いるようにしたので、例えば、暗号化鍵Kが第三者に漏れたとしても、セッション毎に暗号化鍵Kが変化することとなるので、漏れた暗号化鍵Kを用いて生成されたワンタイムID以外のワンタイムIDを予測できなくなる。すなわち、盗聴が困難で安全性に優れたワンタイムIDを生成することが可能になり、ワンタイムIDの将来にわたる安全性(PFS)を実現することが可能になる。

また、上記ワンタイムID (SIGNAL) を用いて、クライアント・サーバ間における認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムIDを識別情報として把握することができる。したがって、DoS攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

また、この実施の形態では、クライアント20の正当性を判定するのに用いる一方向関数 $F_c$ として、共有鍵、DH公開値 $g^*$ 、ID $c$  (クライアントID) およびSIGNALを引数とする疑似乱数関数を用いるとともに、サーバ10の正当性を判定するのに用いる一方向関数 $F_s$ として、共有鍵、DH公開値 $g^*$ 、 $g^*$ 、ID $s$  (サーバID) およびSIGNALを引数とする疑似乱数関数を用いるようにしたので、従来の鍵交換・認証方式では3回必要であった通信回数を2回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

#### (実施の形態3)

前述した第2の実施の形態では、前回のセッションで生成された暗号化鍵 (可変共有鍵) を引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのワンタイムID (SIGNAL) として用いるようにしたが、この第3の実施の形態では、前回のセッションで生成された共有鍵と、当該セッションにおける通信順序とを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションの各通信時におけるワンタイムIDとして用いるようにしている。この第3の実施の形態特有の部分以外は、第2の実施の形態におけると同様である。この第3の実施の形態において、第2の実施の形態と同一部分には同一符号を付し、その説明を省略する。

図9は本発明に係る認証方法の第3の実施の形態を説明する図である。この第3の実施の形態では、まず、ステップP1において、クライアント20が、SIGNAL $n, 1$  (第一のワンタイムID) を生成するとともに、ID $c$  (クライ

アントID)、ID<sub>s</sub> (サーバID)、DH公開値 $g_{xn}$ およびSIGNAL<sub>n, i</sub>を共有鍵 $K_{i-1}$  (第一の可変共有鍵)で暗号化し、この暗号化データとSIGNAL<sub>n, i</sub>とをサーバ10に対して送信する処理を実行する。

ここで、SIGNALは、i番目のセッションにおけるクライアント20のj番目の通信で利用するSIGNALをSIGNAL<sub>i, j</sub>、i番目のセッションにおけるサーバ10のj番目の通信で利用するSIGNALをSIGNAL'<sub>i, j</sub>とした場合、次のように生成される。

$$\begin{aligned} \text{SIGNAL}_{1,j} &= \text{hash}(R, j) & i=1 \\ \text{SIGNAL}_{i,j} &= \text{hash}(K_{i-1}, j) & i \geq 2 \\ \text{SIGNAL}'_{1,j} &= \text{hash}'(R, j) & i=1 \\ \text{SIGNAL}'_{i,j} &= \text{hash}'(K_{i-1}, j) & i \geq 2 \\ & \dots \dots \dots \text{(式6)} \end{aligned}$$

上記SIGNALの定義式(6)において、hashとhash'は互いに異なるハッシュ関数、RはID発行管理サーバ30からサーバ10とクライアント20の双方に発行されて両者間で共有化された乱数、 $K_i$ はi番目のセッションで共有したDH共通鍵 $g^{x_i y_i}$  (共有鍵)である。

すなわち、上記SIGNALの定義式(6)によれば、前回のセッションで生成された共有鍵 $K_{i-1}$ と今回のセッションにおける通信順序jとを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのj番目の通信に用いるSIGNALとしている。ただし、最初のセッション( $i=1$ )では、サーバ・クライアント間で予め共有化された乱数Rと当該セッションにおける通信順序jとを引数とするハッシュ関数の関数値を求め、この関数値を最初のセッションのj番目の通信に用いるSIGNALとしている。

次いで、ステップP2では、サーバ10が、SIGNAL<sub>n, i</sub>を演算により求め、この演算結果と、クライアント20から受信したSIGNAL<sub>n, i</sub>との照合により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵 $K_{i-1}$ を用いて暗号化データを復号し、この復号したデータに含まれる、ID<sub>c</sub>、ID<sub>s</sub>およびSIGNAL<sub>n, i</sub>に基づいて、

クライアント 20 の正当性を判定する処理を実行する。

上記判定の結果、受信データと、サーバ 10 に予め格納された記憶データとが一致して、クライアント 20 が正当であると判定される場合には、前述した SIGNAL の定義式に従って  $SIGNAL'_{n,1}$  (第二のワンタイム ID) を生成するとともに、クライアント 20 から受信した DH 公開値  $g^{x_n}$  と当該サーバ 10 に予め記憶された DH 公開値  $g^{y_n}$  とから DH 共通鍵  $g^{x_n y_n}$  を共有鍵  $K_n$  (第二の可変共有鍵) として生成し、この共有鍵  $K_n$ 、IDc、ID s および  $SIGNAL'_{n,1}$  を引数とするハッシュ関数  $h$  の関数値と、DH 公開値  $g^{y_n}$  と、 $SIGNAL'_{n,1}$  とをクライアント 20 に対して送信する処理を実行する (ステップ P3)。一方、受信データと記憶データとが一致せず、クライアント 20 が正当でないと判定される場合には、クライアント 20 からのアクセスを拒否して、当該認証処理を終了する。

次いで、ステップ P4 では、クライアント 20 が、 $SIGNAL'_{n,1}$  を演算により求め、この演算結果と、サーバ 10 から受信した  $SIGNAL'_{n,1}$  との照合により、サーバ 10 を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ 10 から受信した DH 公開値  $g^{y_n}$  と当該クライアント 20 に予め記憶された DH 公開値  $g^{x_n}$  とから DH 共通鍵  $g^{x_n y_n}$  を共有鍵  $K_n$  として生成するとともに、この共有鍵  $K_n$  を用いてハッシュ関数  $h$  の関数値を演算により求め、この演算結果と、サーバ 10 から受信したハッシュ関数  $h$  の関数値との照合により、サーバ 10 の正当性を判定する処理を実行する。

上記判定の結果、受信データと演算結果とが一致して、サーバ 10 が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、サーバ 10 が正当でないと判定される場合には、サーバ 10 へのアクセスを中止して、当該認証処理を終了する。

なお、クライアント 20 が共有鍵  $K_n$  を共有したことをサーバ 10 側で確認する必要がある場合には、このステップ P4 でクライアント 20 がサーバ 10 の正当性を判定した後に、共有鍵  $K_n$ 、IDc、ID s を引数とするハッシュ関数  $h$

の関数値をサーバ10に対して送信するようにすればよい。

以上のように、この第3の実施の形態によれば、前回のセッションで生成された共有鍵 $K_{i-1}$ （可変共有鍵）と今回のセッションにおける通信順序 $j$ とを引数とするハッシュ関数の関数値を求め、この関数値を当該セッションの $j$ 番目の通信にのみ有効なワнтаイムID（ $SIGNAL$ ）として用いるようにしたので、例えば、 $n$ 番目のセッションで生成した共有鍵 $K_n$ が第三者に漏れたとしても、セッション毎に共有鍵 $K_n$ が変化することとなるので、漏れた共有鍵 $K_n$ を用いて生成されたワнтаイムID（ $SIGNAL_{n+1} \dots j$ 、 $SIGNAL'_{n+1} \dots j$ ）

）以外のワнтаイムIDを予測できなくなる。すなわち、盗聴が困難で安全性に優れたワнтаイムIDを生成することが可能になり、ワнтаイムIDの将来にわたる安全性（ $PFS$ ）を実現することが可能になる。

また、上記ワнтаイムID（ $SIGNAL$ ）を用いて、クライアント・サーバ間における認証を行うようにしたので、前述した第2の実施の形態と同様、大量の計算要求・応答要求などによる計算量やメモリへのDoS攻撃を防止することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。

なお、DoS攻撃を防止する手法の一つとして、クッキー（乱数）を用いた手法が一般に知られている。この方法によれば、IPアドレスとクッキー生成者しか知らない秘密を組み合わせることにより、同一IPアドレスからのDoS攻撃を防ぐことができる。これに対して、本実施の形態の $SIGNAL$ の場合には、DH共通鍵を知らない限り、次回有効となる $SIGNAL$ を予測することができない。よって、毎回の通信に $SIGNAL$ を利用することにより、クッキーと同様の効果が得られる。さらに、クッキーの場合はセッション中にIPアドレスが変わることを許さないが、 $SIGNAL$ は変わっても良い。また、クッキーを用いた場合IPアドレスを偽造したDoS攻撃を防ぐことができないが、ワнтаイムIDではIPアドレスが関係ないためこのような攻撃も防ぐことができる。

また、本実施の形態において、例えば、クライアント20がプロトコルの最初のメッセージを送り（ステップP1）、サーバ10がそれに対応してDH鍵交換

の計算を行い（ステップP2）、2番目のメッセージを送った（ステップP3）場合を考える。もし、サーバ10のメッセージが途中で消失、もしくは攻撃者に横取りされ、クライアント20が受け取ることができなかった場合、クライアント20はもう一度最初のメッセージを送信する必要がある。このとき、サーバ10は正しいクライアント20が通信を送りなおしてきたのか、攻撃者が最初のメッセージを読み取りリプレイ攻撃を行っているのか、判断することができない。そこで、クライアント20はもう一度最初のメッセージを送りなおす場合、最初のチャレンジの際に送ったメッセージと同一の内容のものを送ることにし、サーバ10も以前返信したメッセージのコピーをそのまま送ることにする。これにより、無駄なDH鍵交換の計算を避けることができ、リプレイ攻撃によるDOS攻撃を防ぐことができる。

なお、この実施の形態では、前回のセッションで生成された共有鍵（DH共通鍵） $K_{i-1}$  と今回のセッションにおける通信順序  $j$  とを引数とするハッシュ関数の関数値を求め、この関数値を当該セッションの  $j$  番目の通信にのみ有効なワンタイムID（SIGNAL）として生成するようにしたが、例えば、次のように SIGNAL を生成することも可能である。

$$\begin{aligned} SS_i &= h1(K_{i-1}) \\ SIGNAL_{i,j} &= hash(SS_i, j) \\ SIGNAL'_{i,j} &= hash'(SS_i, j) \\ &\dots\dots\dots (式7) \end{aligned}$$

上記 SIGNAL の定義式（7）において、 $SS_i$  は  $(i-1)$  番目のセッションで共有したDH共通鍵  $K_{i-1}$  を引数とするハッシュ関数の関数値である。

また、この場合には、 $i$  番目のセッションで用いられる認証用鍵を  $AK_i$ 、暗号化鍵を  $SK_i$  として、これら鍵を、例えば、

$$AK_i = h2(K_{i-1}), SK_i = h3(K_{i-1})$$

という式から求めるようにしてもよい。なお、 $h1$ 、 $h2$ 、 $h3$  は、衝突のない一方向性ハッシュ関数である。

このように  $SS_i$  から認証用鍵および暗号化鍵を生成する場合には、前述した

ステップP1において、クライアント20が、 $ID_c$ 、 $ID_s$ 、DH公開値 $g^{x_n}$ および $SIGNAL_{n,1}$ を暗号化してサーバ10に対して送信する際に、認証鍵 $AK_n$ を用いるようにする。また、ステップP3において、サーバ10がクライアント20に対して送信するハッシュ関数 $h$ に、暗号化鍵 $SK_n$ 、 $ID_c$ 、 $ID_s$ および $SIGNAL'_{n,1}$ を引数とするハッシュ関数を用いるようにする。

そうすることで、攻撃者が、仮に、 $SS_i$ 、 $AK_i$ 、 $SK_i$ の何れか1つの値を知ることができたとしても、その他の値を計算することはできない。よって、攻撃者が $i$ 番目のセッションにおいて正規ユーザに成りすまし、鍵交換を行うためには、 $AK_i$ 、 $SIGNAL$ 、正規ユーザのID情報( $ID_s$ 、 $ID_c$ )が必要となり、暗号通信するためには、 $SK_i$ 、 $SIGNAL$ 、正規ユーザのID情報、通信回数が必要となる。

また、 $n$ 番目のセッションにおけるクライアント20のDH公開値 $g^{x_n}$ は、認証鍵 $AK_i$  ( $h_2(K_{i-1})$ )を用いて暗号化される。よって、 $AK_i$ を知らない攻撃者は $g^{x_n}$ を知ることができない。そのため、本方式で生成・共有されるDiffie-Hellman共通鍵は計算量的、かつ情報論的に安全である。

#### (実施の形態4)

前述した第2の実施の形態および第3の実施の形態では、認証と同時にDiffie-Hellman鍵交換を行うようにしたが、この第4の実施の形態では、Diffie-Hellman鍵交換を省略するようにしている。この第4の実施の形態特有の部分以外は、第2の実施の形態におけると同様である。この第4の実施の形態において、第2の実施の形態と同一部分には同一符号を付し、その説明を省略する。

図10は本発明に係る認証方法の第4の実施の形態を説明する図である。この第4の実施の形態では、まず、クライアント20が、乱数 $R_c$  (第一の乱数)を生成するとともに、サーバ10との間で予め共有化された共有鍵 $K_1$  (第一の共有鍵)および乱数 $R_0$  (初期乱数)を引数とする疑似乱数関数 $prf(K_1, R_0)$ の関数値を $SIGNAL_{c,1}$  (第一のワンタイムID)として求め(ステップS11)、この $SIGNAL_{c,1}$ と、共有鍵 $K_1$ で暗号化した乱数 $R_c$ とをサーバ

10に対して送信する処理を実行する（ステップS12）。

次いで、サーバ10が、乱数 $R_s$ （第二の乱数）を生成するとともに、共有鍵 $K_1$ で復号した乱数 $R_c$ と共有鍵 $K_1$ とを引数とする疑似乱数関数 $\text{prf}(K_1, R_c)$ の関数値を $\text{SIGNAL}_{.1}$ （第二のワンタイムID）として求め（ステップS13）、この $\text{SIGNAL}_{.1}$ と、共有鍵 $K_1$ で暗号化した乱数 $R_s$ と、乱数 $R_0 + R_c$ （乱数 $R_0$ 、 $R_c$ を引数とする所定の演算結果、例えば、両者の排他的論理和など）とをクライアント20に対して送信する処理を実行する（ステップS14）。

次いで、クライアント20が、乱数 $R_c$ と共有鍵 $K_1$ に基づいて $\text{SIGNAL}_{.1}$ を演算により求め、この演算結果とサーバ10から受信した $\text{SIGNAL}_{.1}$ との比較により、サーバ10を識別するとともに、乱数 $R_0 + R_c$ の受信データと演算結果との比較により、サーバ10の正当性を判定する処理を実行する（ステップS15）。

上記判定の結果、各々の受信データと演算結果とが一致して、サーバ10が正当であると判定される場合には、クライアント20が、乱数 $R_c$ および乱数 $R_s$ に基づいて共有鍵 $K_2$ （第二の共有鍵）を生成するとともに、この共有鍵 $K_2$ 、乱数 $R_s$ および乱数 $R_c$ を引数とする疑似乱数関数 $\text{prf}(K_2, R_s, R_c)$ の関数値を $\text{SIGNAL}_{.2}$ （第三のワンタイムID）として求め、この $\text{SIGNAL}_{.2}$ と、乱数 $R_c + R_s$ （乱数 $R_c$ 、 $R_s$ を引数とする所定の演算結果）とをサーバ10に対して送信する処理を実行する（ステップS16）。一方、受信データと演算結果とが一致せず、サーバ10が正当でないと判定される場合には、サーバ10へのアクセスを中止して、当該認証処理を終了する。

サーバ10は、クライアント20から $\text{SIGNAL}_{.2}$ を受信すると、乱数 $R_c$ および乱数 $R_s$ に基づいて共有鍵 $K_2$ を生成するとともに、共有鍵 $K_2$ 、乱数 $R_s$ および乱数 $R_c$ に基づいて $\text{SIGNAL}_{.2}$ を演算により求め、この演算結果とクライアント20から受信した $\text{SIGNAL}_{.2}$ との比較により、クライアント20を識別するとともに、乱数 $R_c + R_s$ の受信データと演算結果との比較により、クライアント20の正当性を判定する処理を実行する（ステップS17）。



上記判定の結果、各々の受信データと演算結果とが一致して、クライアント20が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第4の実施の形態によれば、相互認証の過程で生成された乱数と、相互認証の過程で変化する共有鍵 $K$ とを引数とする疑似乱数関数 $\text{prf}$ の関数値をワンタイムIDとして用いるようにしたので、前述した第2の実施の形態と同様、ワンタイムIDの安全性を高めることができ、迅速かつ安全な相互認証を実現することができる。

#### (実施の形態5)

前述した第4の実施の形態では、ワンタイムID (SIGNAL) の生成に用いる共有鍵を相互認証の過程で変化させるようにしたが、この第5の実施の形態では、上記共有鍵を固定するようにしている。

すなわち、この第5の実施の形態では、図11に示すように、先ず、クライアント20が、乱数 $R_c$  (第一の乱数) を生成するとともに、サーバ10との間で予め共有化された共有鍵 $K$ および乱数 $R_0$  (初期乱数) を引数とする疑似乱数関数 $\text{prf}(K, R_0)$ の関数値を $\text{SIGNAL}_{c1}$  (第一のワンタイムID) として求め (ステップS21)、この $\text{SIGNAL}_{c1}$ と、共有鍵 $K$ で暗号化した乱数 $R_c$ とをサーバ10に対して送信する処理を実行する (ステップS22)。

次いで、サーバ10が、乱数 $R_s$  (第二の乱数) を生成するとともに、共有鍵 $K$ で復号した乱数 $R_c$ と共有鍵 $K$ とを引数とする疑似乱数関数 $\text{prf}(K, R_c)$ の関数値を $\text{SIGNAL}_{s1}$  (第二のワンタイムID) として求め (ステップS23)、この $\text{SIGNAL}_{s1}$ と、共有鍵 $K$ で暗号化した乱数 $R_s$ と、乱数 $R_0 + R_c$  (乱数 $R_0$ 、 $R_c$ を引数とする所定の演算結果) とをクライアント20に対して送信する処理を実行する (ステップS24)。

次いで、クライアント20が、乱数 $R_c$ および共有鍵 $K$ に基づいて $\text{SIGNAL}_{c1}$ を演算により求め、この演算結果とサーバ10から受信した $\text{SIGNAL}_{s1}$

との比較により、サーバ10を識別するとともに、乱数 $R_0 + R_c$ の受信データと演算結果との比較により、サーバ10の正当性を判定する処理を実行する（ステップS25）。

上記判定の結果、各々の受信データと演算結果とが一致して、サーバ10が正当であると判定される場合には、クライアント20が、乱数 $R_c$ 、乱数 $R_s$ および共有鍵 $K$ を引数とする疑似乱数関数 $\text{prf}(K, R_s, R_c)$ の関数値を $\text{SIGNAL}_{c2}$ （第三のワンタイムID）として求め、この $\text{SIGNAL}_{c2}$ と、乱数 $R_c + R_s$ （乱数 $R_c$ 、 $R_s$ を引数とする所定の演算結果）とをサーバ10に対して送信する処理を実行する（ステップS26）。一方、受信データと演算結果とが一致せず、サーバ10が正当でないと判定される場合には、サーバ10へのアクセスを中止して、当該認証処理を終了する。

サーバ10は、クライアント20から $\text{SIGNAL}_{c2}$ を受信すると、乱数 $R_c$ 、乱数 $R_s$ および共有鍵 $K$ に基づいて $\text{SIGNAL}_{c2}$ を演算により求め、この演算結果とクライアント20から受信した $\text{SIGNAL}_{c2}$ との比較により、クライアント20を識別するとともに、乱数 $R_c + R_s$ の受信データと演算結果との比較により、クライアント20の正当性を判定する処理を実行する（ステップS27）。

上記判定の結果、各々の受信データと演算結果とが一致して、クライアント20が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第5の実施の形態によれば、相互認証の過程で生成された乱数と共有鍵 $K$ とを引数とする疑似乱数関数 $\text{prf}$ の関数値をワンタイムIDとして用いるようにしたため、例えば、共有鍵 $K$ が第三者に漏れたとしても、乱数によって疑似乱数関数 $\text{prf}$ の関数値が相互認証の過程で順次変化することとなるので、相互認証の過程で生成される乱数がわからない限り、ワンタイムIDを予測することができない。したがって、前述した第2～第4の実施の形態と同様、

ワンタイムIDの安全性を高めることができ、迅速かつ安全な相互認証を実現することができる。

(実施の形態6)

図12は本発明に係る認証方法の第6の実施の形態を説明する図である。この第6の実施の形態では、まず、クライアント20が、乱数 $R_{c,i}$ （第一の乱数）を生成するとともに、サーバ10との間で予め共有化された共有鍵 $K_i$ 、乱数 $R_{c,i-1}$ （第一の記憶乱数）および乱数 $R_{s,i-1}$ （第二の記憶乱数）を引数とする疑似乱数関数 $prf(K_i, R_{c,i-1}, R_{s,i-1})$ の関数値を $SIGNAL_{c,i}$ （第一のワンタイムID）として求める処理を実行する（ステップS31）。

なお、 $R_{c,i}$ は $i$ 番目のセッションでクライアント20により生成された乱数、 $R_{s,i}$ は $i$ 番目のセッションでサーバ10により生成された乱数、 $K_i$ は $i$ 番目のセッションで使用する可変共有鍵をそれぞれ示している。また、前回（ $i-1$ 番目）のセッションで生成された乱数 $R_{c,i-1}$ 、 $R_{s,i-1}$ は、サーバ10とクライアント20の各記憶装置13、23の記憶領域に格納されており、これら乱数 $R_{c,i-1}$ 、 $R_{s,i-1}$ に基づいて、共有鍵 $K_i$ が生成されるようになっている。

そして、クライアント20は、 $SIGNAL_{c,i}$ を生成した後、 $ID_c$ （クライアントID）、 $ID_s$ （サーバID）および乱数 $R_{c,i}$ を共有鍵 $K_i$ で暗号化した暗号化データ $E_{K_i}(ID_c, ID_s, R_{c,i})$ と、 $SIGNAL_{c,i}$ とをサーバ10に対して送信する処理を実行する（ステップS32）。

サーバ10は、クライアント20から $SIGNAL_{c,i}$ を受信すると、共有鍵 $K_i$ 、乱数 $R_{c,i-1}$ および乱数 $R_{s,i-1}$ に基づいて $SIGNAL_{c,i}$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_{c,i}$ との比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵 $K_i$ を用いて暗号化データ $E_{K_i}(ID_c, ID_s, R_{c,i})$ を復号し、この復号したデータに含まれる $ID_c$ および $ID_s$ に基づいて、クライアント20の正当性を判定する処理を実行する。

上記判定の結果、受信データと、サーバ10に予め格納された記憶データとが一致して、クライアント20が正当であると判定される場合には、乱数 $R_{s,i}$ （第

二の乱数)を生成するとともに、乱数 $R_{c,i}$ 、乱数 $R_{s,i-1}$ および共有鍵 $K_i$ を引数とする疑似乱数関数 $prf(K_i, R_{c,i}, R_{s,i-1})$ の関数値を $SIGNAL_{s,i}$ (第二のワнтаイムID)として求める。そして、乱数 $R_{c,i-1}$ 、 $R_{s,i-1}$ を格納していた記憶領域に、乱数 $R_{c,i}$ 、 $R_{s,i}$ をそれぞれ格納するとともに、これら乱数 $R_{c,i}$ 、 $R_{s,i}$ に基づき共有鍵 $K_{i+1}$ を生成して記憶する処理を実行する(ステップS33)。

次いで、サーバ10は、IDc、IDsおよび乱数 $R_{s,i}$ を共有鍵 $K_i$ で暗号化した暗号化データ $E_{x,i}(ID_s, ID_c, R_{s,i})$ と、 $SIGNAL_{s,i}$ とをクライアント20に対して送信する処理を実行する(ステップS34)。

一方、受信データと記憶データとが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

クライアント20は、サーバ10から $SIGNAL_{s,i}$ を受信すると、共有鍵 $K_i$ 、乱数 $R_{c,i}$ および乱数 $R_{s,i-1}$ に基づいて $SIGNAL_{s,i}$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_{s,i}$ との比較により、サーバ10を識別し、識別できない場合には、通信を拒否する。一方、識別できる場合には、共有鍵 $K_i$ を用いて暗号化データ $E_{x,i}(ID_s, ID_c, R_{s,i})$ を復号し、この復号したデータに含まれるIDcおよびIDsに基づいて、サーバ10の正当性を判定する処理を実行する。サーバ10を識別できる場合、通信相手を特定できるだけでなく、サーバ10が乱数 $R_{c,i}$ を受け取ったことも確認することができる。

そして、上記判定の結果、受信データと、クライアント20に予め格納された記憶データとが一致して、サーバ10が正当であると判定される場合には、乱数 $R_{c,i-1}$ 、 $R_{s,i-1}$ を格納していた記憶領域に、乱数 $R_{c,i}$ 、 $R_{s,i}$ をそれぞれ格納して、これら乱数 $R_{c,i}$ 、 $R_{s,i}$ に基づき共有鍵 $K_{i+1}$ を生成・記憶した後(ステップS35)、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと記憶データとが一致せず、サーバ10が正当でないと判定される場合には、サーバ10からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第6の実施の形態によれば、前述した第4の実施の形態と同様の作用・効果が得られるのに加えて、 $ID_c$ 、 $ID_s$ および乱数 $R_{s,i}$ を共有鍵 $K_i$ で暗号化した暗号化データ $E_{K_i}(ID_s, ID_c, R_{s,i})$ を通信相手に送信するようにしたことにより、例えば、攻撃者によって暗号化データが書き換えられた場合においても、暗号化データに含まれるID情報( $ID_s$ 、 $ID_c$ )が正しく復号されないために、このデータを受け取ったサーバ10またはクライアント20は、送られてきた暗号化データが誤ったものであることを容易に検出でき、乱数を受け取らずに廃棄することが可能となる。また、 $SIGNAL_{c,i}$ の値が他の複数のクライアントと重複した場合においても、暗号化データに含まれるID情報( $ID_s$ 、 $ID_c$ )を参照することによって、通信相手を容易に特定することができる。

さらに、この第6の実施の形態によれば、通信相手がサーバ・クライアントのID情報( $ID_s$ 、 $ID_c$ )を正しく暗号化できているか否かを確認することによって、通信相手の正当性を判定するようにしたので、前述した第4の実施の形態では3回必要であった通信回数を2回に低減することが可能となり、より効率的な認証が可能となる。

#### (実施の形態7)

図13は本発明に係る認証方法の第7の実施の形態を説明する図である。この第7の実施の形態では、まず、クライアント20が、乱数 $R_{c,i}$ (第一の乱数)を生成するとともに、サーバ10との間で予め共有化された固定共有鍵 $K$ 、乱数 $R_{c,i-1}$ (第一の記憶乱数)および乱数 $R_{s,i-1}$ (第二の記憶乱数)を引数とする疑似乱数関数 $prf(K, R_{c,i-1}, R_{s,i-1})$ の関数値を $SIGNAL_{c,i}$ (第一のワンタイムID)として求める処理を実行する(ステップS41)。

なお、 $R_{c,i}$ は*i*番目のセッションでクライアント20により生成された乱数、 $R_{s,i}$ は*i*番目のセッションでサーバ10により生成された乱数をそれぞれ示している。また、前回(*i*-1番目)のセッションで生成された乱数 $R_{c,i-1}$ 、 $R_{s,i-1}$ は、サーバ10とクライアント20の各記憶装置13、23の記憶領域に格納されている。

そして、クライアント20は、 $SIGNAL_{c1}$ を生成した後、 $ID_c$ （クライアントID）、 $ID_s$ （サーバID）および乱数 $R_{c1}$ を共有鍵 $K$ で暗号化した暗号化データ $E_K$ （ $ID_c$ 、 $ID_s$ 、 $R_{c1}$ ）と、 $SIGNAL_{c1}$ とをサーバ10に対して送信する処理を実行する（ステップS42）。

サーバ10は、クライアント20から $SIGNAL_{c1}$ を受信すると、共有鍵 $K$ 、乱数 $R_{c1-1}$ および乱数 $R_{s1-1}$ に基づいて $SIGNAL_{c1}$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_{c1}$ との比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵 $K$ を用いて暗号化データ $E_K$ （ $ID_c$ 、 $ID_s$ 、 $R_{c1}$ ）を復号し、この復号したデータに含まれる $ID_c$ および $ID_s$ に基づいて、クライアント20の正当性を判定する処理を実行する。

上記判定の結果、受信データと、サーバ10に予め格納された記憶データとが一致して、クライアント20が正当であると判定される場合には、乱数 $R_{s1}$ （第二の乱数）を生成するとともに、乱数 $R_{c1}$ 、乱数 $R_{s1-1}$ および共有鍵 $K$ を引数とする疑似乱数関数 $prf(K, R_{c1}, R_{s1-1})$ の関数値を $SIGNAL_{s1}$ （第二のワンタイムID）として求める。そして、乱数 $R_{c1-1}$ 、 $R_{s1-1}$ を格納していた記憶領域に、乱数 $R_{c1}$ 、 $R_{s1}$ をそれぞれ格納する処理を実行する（ステップS43）。

次いで、サーバ10は、 $ID_c$ 、 $ID_s$ および乱数 $R_{s1}$ を共有鍵 $K$ で暗号化した暗号化データ $E_K$ （ $ID_s$ 、 $ID_c$ 、 $R_{s1}$ ）と、 $SIGNAL_{s1}$ とをクライアント20に対して送信する処理を実行する（ステップS44）。

一方、受信データと記憶データとが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

クライアント20は、サーバ10から $SIGNAL_{s1}$ を受信すると、共有鍵 $K$ 、乱数 $R_{c1}$ および乱数 $R_{s1-1}$ に基づいて $SIGNAL_{s1}$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_{s1}$ との比較により、サーバ10を識別し、識別できない場合には、通信を拒否する。一方、識別できる場合

には、共有鍵 $K$ を用いて暗号化データ $EK (ID_s, ID_c, R_{s,i})$ を復号し、この復号したデータに含まれる $ID_c$ および $ID_s$ に基づいて、サーバ10の正当性を判定する処理を実行する。サーバ10を識別できる場合、通信相手を特定できるだけでなく、サーバ10が乱数 $R_{s,i}$ を受け取ったことも確認することができる。

そして、上記判定の結果、受信データと、クライアント20に予め格納された記憶データとが一致して、サーバ10が正当であると判定される場合には、乱数 $R_{c,i-1}$ 、 $R_{s,i-1}$ を格納していた記憶領域に、乱数 $R_{c,i}$ 、 $R_{s,i}$ をそれぞれ格納して、これら乱数 $R_{c,i}$ 、 $R_{s,i}$ に基づき共有鍵 $K$ を生成・記憶した後（ステップS45）、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと記憶データとが一致せず、サーバ10が正当でないと判定される場合には、サーバ10からのアクセスを拒否して、当該認証処理を終了する。

以上のように、この第7の実施の形態によれば、前述した第5の実施の形態と同様の作用・効果が得られるのに加えて、例えば、攻撃者によって暗号化データが書き換えられた場合においても、データを受け取ったサーバ10またはクライアント20は、送られてきた暗号化データが誤ったものであることを容易に検出でき、乱数を受け取らずに廃棄することが可能となる。また、 $SIGNAL_{s,i}$ の値が他の複数のクライアントと重複した場合においても、暗号化データに含まれるID情報（ $ID_s$ 、 $ID_c$ ）を参照することによって、通信相手を容易に特定することができる。さらに、この第7の実施の形態によれば、前述した第4の実施の形態では3回必要であった通信回数を2回に低減することが可能となり、より効率的な認証が可能となる。

#### （実施の形態8）

この第8の実施の形態では、ワンタイムIDを用いたリプレイ攻撃の防止方法について説明する。リプレイ攻撃とは、過去に正式な通信者が送信したときに有効であった通信情報を攻撃者（第三者）が盗聴し、再利用する攻撃のことである。

先ず、OSP A (Optimal Strong Password Authentication) と呼ばれるパスワードを利用した認証方式 (Chun-Li LIN, Hung-Min SUN, Tzonelih HWANG, A

ttacks and Solutions on Strong- Password Authentication, IEICE TRANS. COMMUN., VOL.E84-B, NO.9, September 2001.) について、図 1 4 に基づいて説明する。

当該認証に先立って、クライアント 2 0 には、ハッシュ関数  $h$  およびパスワード  $P$  が予め記憶保持されており、サーバ 1 0 には、ハッシュ関数  $h$ 、セッション回数  $n$ 、 $ID_c$  (クライアント  $ID$ ) および検証用情報  $h^2 (P@n)$  が予め記憶保持されている。検証用情報  $h^2 (P@n)$  は、クライアント 2 0 の正当性を検証するための情報であって、パスワード  $P$  と通信回数  $n$  の排他的論理和を用いてハッシュ関数  $h$  により生成された情報である。なお、 $h^2 (P@n)$  は、ハッシュ関数  $h$  の計算を 2 回行うこと、つまり  $h (h (P@n))$  を示しており、この数式中の  $@$  は排他的論理和を示している。

この認証方式では、先ず、クライアント 2 0 がサーバ 1 0 に対して  $ID_c$  を送信する (ステップ S 5 1)。

サーバ 1 0 は、クライアント 2 0 から  $ID_c$  を受信すると、この受信した  $ID_c$  と、予め記憶している  $ID_c$  との比較により、クライアント 2 0 を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ 1 0 に対してセッション回数  $n$  を送信する (ステップ S 5 2)。

クライアント 2 0 は、サーバ 1 0 からセッション回数  $n$  を受信すると、この受信したセッション回数  $n$ 、予め記憶しているハッシュ関数  $h$  およびパスワード  $P$  を用いて、第 1 ~ 第 3 の認証用情報  $C_1$ 、 $C_2$ 、 $C_3$  を生成し (ステップ S 5 3)、これら  $C_1$ 、 $C_2$ 、 $C_3$  をサーバ 1 0 に対して送信する (ステップ S 5 4)。ここで、 $C_1 = h (P@n) @ h^2 (P@n)$ 、 $C_2 = h^2 (P@ (n+1)) @ h (P@n)$ 、 $C_3 = h^3 (P@ (n+1))$  である。

サーバ 1 0 は、クライアント 2 0 から  $C_1$ 、 $C_2$ 、 $C_3$  を受信すると、先ず、受信した  $C_1 \neq C_2$  であることを確認する。これは、 $C_1 = h (P@n) @ h^2 (P@n)$ 、 $C_2 = h (P@n) @ h^2 (P@n)$ 、 $C_3 = h^3 (P@n)$  と計算して送った場合においても、サーバ 1 0 がクライアント 2 0 を認証してしまい、次の検証用情報として、 $h^2 (P@ (n+1))$  ではなく  $h^2 (P@n)$  を記憶



してしまう不具合が発生する可能性があることから、このような不具合の発生を防ぐために行われるものである。

次いで、サーバ10は、C1、C2から、 $h(P@n)$ 、 $h^2(P@(n+1))$ を演算により求める。すなわち、受信したC1と、予め記憶している検証用情報 $h^2(P@n)$ との排他的論理和を求めることで $h(P@n)$ を導き出し、この $h(P@n)$ と受信したC2との排他的論理和を求めることで $h^2(P@(n+1))$ を導き出す。

次いで、予め記憶しているハッシュ関数 $h$ を用いて、求めた $h(P@n)$ から $h(h(P@n))$ を計算し、この $h(h(P@n))$ が、予め記憶している検証用情報 $h^2(P@n)$ と一致するか否かを検証する。同時に、求めた $h^2(P@(n+1))$ から上記ハッシュ関数 $h$ を用いて $h(h^2(P@(n+1)))$ を計算し、この $h(h^2(P@(n+1)))$ が、受信したC3と一致するか否かを検証する（ステップS55）。

これら検証の結果、何れもが一致して、クライアント20が正当であると判定される場合には、検証用情報を $h^2(P@n)$ から $h^2(P@(n+1))$ に更新し、セッション回数を $n$ から $n+1$ に更新した後、クライアント20からのアクセスを承諾して、当該認証処理を終了する。一方、上記検証の結果、少なくとも何れか一方が一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

上記認証方式によれば、盗聴者に対して安全な認証を行うことができるとともに、セッション毎に検証用情報を $h^2(P@n)$ から $h^2(P@(n+1))$ へと更新することができるという利点がある。

しかしながら、上記認証方式にあっては、一度使用された認証情報C1、C2、C3をもう一度利用することによるリプレイ攻撃を防止することができないという問題点があった。

そこで、本発明者等は、このような問題点を解決する認証方式として、次のような認証方式を開発した。

図15は、本発明に係る認証方法の第8の実施の形態を説明する図である。こ

の図15に示すように、クライアント20に、ハッシュ関数 $h$ およびパスワード $P$ が予め記憶保持され、サーバ10に、ハッシュ関数 $h$ 、セッション回数 $n$ 、 $ID_c$ および検証用情報 $h^2(P@n)$ が予め記憶保持されている場合には、先ず、クライアント20がサーバ10に対して $ID_c$ を送信する(ステップS61)。

サーバ10は、クライアント20から $ID_c$ を受信すると、この受信した $ID_c$ と、予め記憶している $ID_c$ との比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ10に対してセッション回数 $n$ を送信する(ステップS62)。

クライアント20は、サーバ10からセッション回数 $n$ を受信すると、この受信したセッション回数 $n$ 、予め記憶しているハッシュ関数 $h$ およびパスワード $P$ を用いて、第1～第3の認証用情報 $C_1$ 、 $C_2$ 、 $C_3$ 、 $SIGNAL_n$ を生成し(ステップS63)、これら $C_1$ 、 $C_2$ 、 $C_3$ 、 $SIGNAL_n$ をサーバ10に対して送信する(ステップS64)。ここで、 $C_1 = h(P@n) @ h^2(P@n)$ 、 $C_2 = h^2(P@(n+1)) @ h(P@n)$ 、 $C_3 = h^3(P@(n+1))$ 、 $SIGNAL_n = h(h^2(P@n), n)$ である。即ち、 $n$ 番目のセッションで使用するワンタイムIDである $SIGNAL_n$ が、検証用情報 $h^2(P@n)$ およびセッション回数 $n$ を引数とするハッシュ関数 $h$ の関数値となっている。

サーバ10は、クライアント20から $C_1$ 、 $C_2$ 、 $C_3$ 、 $SIGNAL_n$ を受信すると、先ず、予め記憶している検証用情報 $h^2(P@n)$ とセッション回数 $n$ とに基づいて $SIGNAL_n$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_n$ との比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、受信した $C_1 \neq C_2$ であることを確認した後、 $C_1$ および $C_2$ から $h(P@n)$ 、 $h^2(P@(n+1))$ を演算により求める。

次いで、サーバ10は、予め記憶しているハッシュ関数 $h$ を用いて、求めた $h(P@n)$ から $h(h(P@n))$ を計算し、この $h(h(P@n))$ が、予め記憶している検証用情報 $h^2(P@n)$ と一致するか否かを検証する。同時に、

求めた  $h^2(P@n+1)$  から上記ハッシュ関数  $h$  を用いて  $h(h^2(P@n+1))$  を計算し、この  $h(h^2(P@n+1))$  が、受信した  $C3$  と一致するか否かを検証する（ステップ  $S65$ ）。

これら検証の結果、何れもが一致して、クライアント  $20$  が正当であると判定される場合には、検証用情報を  $h^2(P@n)$  から  $h^2(P@n+1)$  に更新し、セッション回数を  $n$  から  $n+1$  に更新した後、クライアント  $20$  からのアクセスを承諾して、当該認証処理を終了する。一方、上記検証の結果、少なくとも何れか一方が一致せず、クライアント  $20$  が正当でないと判定される場合には、クライアント  $20$  からのアクセスを拒否して、当該認証処理を終了する。

上記認証方式によれば、検証用情報である  $h^2(P@n)$  が攻撃者に知られる虞がないので、次のセッションの  $SIGNAL$  が攻撃者に予測されることはない。しかも、 $SIGNAL$  は他のセッションで使用できないので、攻撃者によるリプレイ攻撃を効果的に防ぐことができる。

なお、図  $16$  に示すように、ハッシュ関数  $h$  およびパスワード  $P$  に加えて、予めセッション回数  $n$  もクライアント  $20$  に記憶保持されている場合には、前述したステップ  $S61$ 、 $S62$  の処理を省略することが可能である。したがって、この場合には、 $ID$  情報 ( $IDc$ ) の盗聴に対する保護を図りつつも、攻撃者によるリプレイ攻撃を効果的に防ぐことが可能である。

なお、以上の各実施の形態においては、複数の装置間の認証にワンタイム  $ID$  を用いるようにしたが、一装置内の複数のアプリケーション間の認証にワンタイム  $ID$  を用いることも可能である。また、以上の各実施の形態においては、本発明に係る認証方法をクライアントサーバシステムに適用した場合について例示したが、これに限られるものではなく、例えば、 $P2P$  (Peer to Peer) システムに本発明に係る認証方法を適用することも可能である。

また、本発明に係る認証方法をユーザによるアクセス毎に利用することも可能であり、その場合には、ユーザによるパスワードの入力を促して、パスワード、若しくはパスワードから生成した値（ワンタイムパスワードを含む。）をワンタイム  $ID$  とともに認証用のデータとして用いることが可能である。

以上説明したように、本発明によれば、盗聴が困難で安全性に優れたワンタイムIDを生成することが可能となり、ワンタイムIDの将来にわたる安全性(PFS)を実現することが可能となる。

また、本発明におけるワンタイムIDの生成方法により生成されたワンタイムIDを用いて、装置間(クライアント・サーバ間)の認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムIDを識別情報として把握できるようになる。

したがって、DoS攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

また、本発明によれば、従来の鍵交換・認証方式では3回必要であった通信回数を2回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

本発明は、図面に示す好ましい実施の形態に基づいて説明されてきたが、当業者であれば、本発明の思想を逸脱することなく容易に各種の変更、改変し得ることは明らかである。本発明はそのような変更例も含むものである。

## 請 求 の 範 囲

1. 通信回線を介して接続された第1認証装置と第2認証装置の相互関係を認証する相互認証方法であって、

前記第1認証装置を特定するための記憶データと、第2認証装置を特定するための記憶データとを、前記第1認証装置及び第2認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを用いて更新した更新結果を履歴データとして、前記第1認証装置及び第2認証装置の各々に共通に記憶する記憶工程と、

前記第1認証装置は、記憶されている履歴データを用いて記憶データを新規に生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第2認証装置に送信する第1送信工程と、前記第2認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第1更新工程と、を含み、

前記第2認証装置は、前記第1認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成しかつ生成した新規の記憶データを前記履歴データを用いて暗号化して第1認証装置に送信する第2送信工程と、前記第1認証装置からの記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第2更新工程とを含み、

前記第1認証装置及び第2認証装置の少なくとも一方の装置において、履歴データに基づいて記憶データの正当性が成立したときに、第1認証装置と第2認証装置の相互関係が正当であると検証することを特徴とする相互認証方法。

2. 前記履歴データを履歴データKとして、該履歴データKとして記憶する、前記第1認証装置を特定するための記憶データは、暗証データC及び認証データRであり、前記第2認証装置を特定するための記憶データは、暗証データS及び認証データQであることを特徴とする請求項1に記載の相互認証方法。

3. 前記第1送信工程は、記憶されている履歴データKの暗証データS及び認証データRを用いて暗証データCを新規に生成しかつ、記憶されている履歴データKの認証データRについて新規に生成し、生成した新規の認証デー

タ R を前記履歴データ K を用いて暗号化して認証データ A を求め、前記認証データ A 及び新規の暗証データ C を第 2 認証装置に送信し、

前記第 1 更新工程は、前記第 2 認証装置からのデータを受信し、前記送信した新規の暗証データ C、受信した新規に生成された暗証データ S、受信した新規に生成された認証データ Q、及び前記送信した新規の認証データ R により、前記履歴データ K を更新し、

前記第 2 送信工程は、前記第 1 認証装置からのデータを受信し、受信した新規の暗証データ C 及び記憶されている履歴データ K の認証データ Q を用いて暗証データ S を新規に生成しかつ記憶されている履歴データ K の認証データ Q について新規に生成し、生成した新規の認証データ Q を記憶した履歴データ K を用いて暗号化して認証データ B を求め、前記認証データ B 及び新規の暗証データ S を第 1 認証装置に送信し、

前記第 2 更新工程は、受信した新規の暗証データ C、新規に生成した暗証データ S、新規に生成した認証データ Q、及び受信した新規の認証データ R により、前記履歴データ K を更新し、

前記第 1 認証装置及び第 2 認証装置の少なくとも一方の装置において、履歴データ K に基づいて暗証データの正当性が成立したときに、第 1 認証装置と第 2 認証装置の相互関係が正当であると検証することを特徴とする請求項 2 に記載の相互認証方法。

4. 前記記憶工程は、前記第 1 送信工程、第 1 更新工程、第 2 送信工程、及び第 2 更新工程における認証による更新結果を履歴データとして記憶することを特徴とする請求項 1 に記載の相互認証方法。

5. 前記認証データ R 及び認証データ Q の少なくとも一方は、乱数発生手段により発生された乱数、データ容量、時間データの少なくとも 1 つであることを特徴とする請求項 2 に記載の相互認証方法。

6. 前記第 1 認証装置の第 1 送信工程では、前記暗証データ S 及び認証データ R による予め定めた関数の演算結果の値を暗証データ C として生成し、前記第 2 認証装置の第 2 送信工程では、前記暗証データ C 及び前記認証データ Q

による予め定めた関数の演算結果の値を暗証データ S として生成することを特徴とする請求項 2 に記載の相互認証方法。

7. 前記第 1 認証装置の第 1 送信工程では、前記生成した新規の認証データ R 及び前記履歴データ K による予め定めた関数の演算結果の値を認証データ A として求め、前記第 2 認証装置の第 2 送信工程では、前記生成した新規の認証データ Q 及び前記履歴データ K による予め定めた関数の演算結果の値を認証データ B として求めることを特徴とする請求項 2 に記載の相互認証方法。

8. 前記第 1 認証装置の検証工程は、前記履歴データ K のうち記憶されている認証データ Q 及び前回送信する前に生成した暗証データ C による予め定めた関数の演算結果の値が受信した暗証データ S と一致するときに前記相互関係が正当であると検証することを特徴とする請求項 2 に記載の相互認証方法。

9. 前記第 2 認証装置の検証工程は、前記履歴データ K のうち記憶されている暗証データ S 及び認証データ R による予め定めた関数の演算結果の値が受信した暗証データ C と一致するときに前記相互関係が正当であると検証することを特徴とする請求項 2 に記載の相互認証方法。

10. 前記記憶工程は、前記第 1 送信工程、第 2 送信工程、第 1 更新工程及び第 2 更新工程を複数実施した結果、得られるデータを履歴データ K として記憶することを特徴とする請求項 2 に記載の相互認証方法。

11. 通信回線を介して接続された第 1 認証装置と第 2 認証装置とから成り、前記第 1 認証装置と第 2 認証装置との間の相互関係を認証する相互認証装置であって、

前記第 1 認証装置に設けられ当該第 1 認証装置を特定するための記憶データを格納する第 1 のメモリと、

前記第 2 認証装置に設けられ当該第 2 認証装置を特定するための記憶データを格納する第 2 のメモリと、

前記第 1 認証装置及び第 2 認証装置の間で予め相互になされた認証による認証毎に前回の認証による記憶データを格納する認証データ記憶手段と、

前記認証データを用いて更新した更新結果を履歴データとして、前記第 1 認証

装置及び第 2 認証装置の各々に共通に記憶する履歴データ記憶手段と、

前記第 1 認証装置又は第 2 認証装置のうち、認証用データ送信側の認証装置に設けられ、前記履歴データを用いて記憶データを新規に生成する記憶データ生成手段と、

生成した新規の記憶データを前記履歴データを用いて暗号化して認証用データ受信側の認証装置に送信する第 1 の送信手段と、

認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び記憶されている履歴データを用いて新規に記憶データを生成する記憶データ生成手段と、

生成した新規の記憶データを前記履歴データを用いて暗号化して前記認証用データ送信側の認証装置に返信する第 2 の送信手段と、

認証用データ送信側の認証装置に設けられ、前記認証用データ受信側の認証装置から返信された記憶データ及び前記送信した新規の記憶データによって前記履歴データを更新する第 1 の更新手段と、

認証用データ受信側の認証装置に設けられ、前記認証用データ送信側の認証装置からの記憶データ及び前記返信した新規の記憶データによって前記履歴データを更新する第 2 の更新手段とを含み、

前記第 1 認証装置及び第 2 認証装置の少なくとも一方の装置において、前記履歴データに基づいて記憶データの正当性が成立したときに、第 1 認証装置と第 2 認証装置の相互関係が正当であると検証する検証手段とを備えた、ことを特徴とする相互認証装置。

1 2. 生成した新規の記憶データを前記履歴データを用いて暗号化するための認証用データを演算する演算手段を有することを特徴とする請求項 1 1 記載の相互認証装置。

1 3. 前記演算手段により認証用データを生成するに際し、暗号化用のデータを生成する乱数発生手段を有することを特徴とする請求項 1 2 記載の相互認証装置。

1 4. 複数の相互認証を行う装置間またはアプリケーション間における認証に



において、一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

15. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

16. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

17. 一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する

る可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

18. 一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

19. 一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

20. 装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイム

IDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、上記第二装置との間で予め共有化された可変共有鍵を用いて上記ワンタイムIDを生成するとともに、この生成したワンタイムIDと、当該第一装置に予め設定されたIDを少なくとも引数とする一方向関数 $F_c$ の関数値と、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方とを上記第二装置に対して送信するステップと、

上記第二装置が、上記ワンタイムIDおよび上記一方向関数 $F_c$ の関数値を演算により求め、この演算結果と、上記第一装置から受信したワンタイムIDおよび一方向関数 $F_c$ の関数値との照合により、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、当該第二装置に予め設定されたIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とを上記第一装置に対して送信するステップと、

上記第一装置が、上記一方向関数 $F_s$ の関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数 $F_s$ の関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とする認証方法。

21. 上記一方向関数 $F_c$ として、所定の共有鍵、上記Diffie-Hellman公開値の一方、上記第一装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いるとともに、

上記一方向関数 $F_s$ として、上記所定の共有鍵、上記Diffie-Hellman公開値の一方、上記Diffie-Hellman公開値の他方、上記第二装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いるようにしたことを特徴とする請求項20に記載の認証方法。

22. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する

情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、上記第二装置との間で予め共有化された第一の可変共有鍵と当該第一装置の通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムIDとして生成するとともに、上記第一の可変共有鍵を用いて、当該第一装置に予め設定されたID、上記第二装置に予め設定されたID、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムIDを暗号化し、この暗号化データと上記第一のワンタイムIDとを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一のワンタイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムIDとの照合により、上記第一装置を識別するステップと、

上記第二装置が、上記第一装置を識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第一のワンタイムIDに基づいて、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、上記第一の可変共有鍵と当該第二装置の通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムIDとして生成するとともに、上記第一装置から受信したDiffie-Hellman公開値の一方と当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二のワンタイムIDを引数とする一方向関数 $h$ の関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記第一装置に対して送信するステップと、

上記第一装置が、上記第二のワンタイムIDを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムIDとの照合により、上記

第二装置を識別するステップと、

上記第一装置が、上記第二装置を識別できた場合に、上記第二装置から受信した上記Diffie-Hellman公開値の他方と当該第一装置に予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数 $h$ の関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数 $h$ の関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とする認証方法。

23. 上記第二のワンタイムIDを生成する一方向関数として、上記第一のワンタイムIDを生成する一方向関数とは異なる一方向関数を用いるようにしたことを特徴とする請求項22に記載の認証方法。

24. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数とを上記第二装置に対して送信するステップと、

上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして求め、この第二のワンタイムIDと上記第二の乱数とを上記第一装置に対して送信するステップと、

上記第一装置が、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムIDとの比較により、上記第二装置の正当性を判定するス

トップと、

上記第一装置が、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムIDとして求め、この第三のワンタイムIDを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とする認証方法。

25. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数を上記第二装置に対して送信するステップと、

上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして求め、この第二のワンタイムIDと上記第二の乱数を上記第一装置に対して送信するステップと、

上記第一装置が、上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムIDとの比較により、上記第二装置の正当性を判定するステップと、

上記第一装置が、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムIDとして求め、この第三のワンタイムIDを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とする認証方法。

26. 上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とする請求項24に記載の認証方法。

27. 上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とする請求項に記載の認証方法。

28. 上記第二装置が上記第二のワンタイムIDと上記第二の乱数とを上記第一装置に対して送信するステップにおいて、上記第二装置は、上記第一装置との間で予め共有化された乱数を初期乱数として、この初期乱数と上記第一の乱数を引数とする所定の演算を行い、この演算結果を上記第一装置に対して送信する一方、上記第一装置は、上記第二装置の正当性の判定材料として、上記第二装置から受信した上記演算結果を、上記第二のワンタイムIDとともに用いることを特徴とする請求項24乃至請求項26の何れかに記載の認証方法。

29. 上記第一装置が上記第三のワンタイムIDを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記第一の乱数と上記第二の乱数を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワンタイムIDとともに用いることを特徴とする請求項24に記載の認証方法。

30. 上記第一装置が上記第三のワンタイムIDを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記第一の乱数と上記第二の乱数

を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワンタイムIDとともに用いることを特徴とする請求項25に記載の認証方法。

31. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、当該第一装置に予め設定されたID、上記第二装置に予め設定されたIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムIDとを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一のワンタイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムIDとの照合により、上記第一装置を識別するステップと、

上記第二装置が、上記第一装置を識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたIDおよび当該第二装置に予め設定されたIDに基づいて、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムIDとして求め、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムID



とを上記第一装置に対して送信するステップと、

上記第二装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップと、

上記第一装置が、上記第二のワンタイムIDを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムIDとの照合により、上記第二装置を識別するステップと、

上記第一装置が、上記第二装置を識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる、上記第二装置に予め設定されたIDおよび当該第一装置に予め設定されたIDに基づいて、上記第二装置の正当性を判定するステップと、

上記第一装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップとを有することを特徴とする認証方法。

32. 上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後に、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようにしたことを特徴とする請求項31に記載の認証方法。

33. 装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で認証を行うサーバであって、

上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムIDとを上記クライアントから受信する受信手段と、

上記一方向関数の関数値 $F_c$ および上記ワンタイムIDを演算により求め、こ

の演算結果と、上記クライアントから受信した上記ワンタイムIDおよび上記一方向関数 $F_c$ の関数値との比較により、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、当該サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する送信手段とを備えることを特徴とするサーバ。

34. 装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてサーバとの間で認証を行うクライアントであって、

上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムIDを生成するとともに、当該クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値を演算により求め、これらワンタイムIDおよび一方向関数 $F_c$ の関数値と、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する送信手段と、

上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、

上記一方向関数 $F_s$ の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数 $F_s$ の関数値との比較により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするクライアント。

35. サーバとクライアントとにより構成された認証システムにおいて、上記サーバおよびクライアントは、装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通

信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で認証を行い、

上記サーバは、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムIDとを上記クライアントから受信する受信手段と、

上記一方向関数の関数値 $F_c$ および上記ワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムIDおよび上記一方向関数 $F_c$ の関数値との比較により、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、当該サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する送信手段とを備え、また、

上記クライアントは、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムIDを生成するとともに、当該クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値を演算により求め、これらワンタイムIDおよび一方向関数 $F_c$ の関数値と、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する送信手段と、

上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、

上記一方向関数 $F_s$ の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数 $F_s$ の関数値との比較により、上記サーバの正当性を判定する判定手段とを備えてなることを特徴とする認証システム。

36. 装置間またはアプリケーション間における認証において、一回限り

使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDに基づいてクライアントとの間で認証を行うサーバに実行させるプログラムであって、

上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムIDとを上記クライアントから受信する処理と、

上記一方向関数の関数値 $F_c$ および上記ワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムIDおよび上記一方向関数 $F_c$ の関数値との比較により、上記クライアントの正当性を判定する処理と、

上記クライアントが正当であると判定された場合に、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 $F_s$ の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する処理とを上記サーバに実行させることを特徴とするプログラム。

37. 装置間またはアプリケーション間における認証において、一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDに基づいてサーバとの間で認証を行うクライアントに実行させるプログラムであって、

上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムIDを生成するとともに、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 $F_c$ の関数値を演算により求め、これらワンタイムIDおよび一方向関数 $F_c$ の関数値と、上記クライアントに予め記憶されたDi

Diffie-Hellman公開値の一方とを上記サーバに対して送信する処理と、

上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数F<sub>s</sub>の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する処理と、

上記一方向関数F<sub>s</sub>の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数F<sub>s</sub>の関数値との比較により、上記サーバの正当性を判定する処理とを上記クライアントに実行させることを特徴とするプログラム。

38. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で認証を行うサーバであって、

上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムIDとして、この第一のワンタイムID、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバID、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワンタイムIDとを上記クライアントから受信する受信手段と、

上記第一のワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントID、上記サーバIDおよび上記第一のワンタイムIDに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の

可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムIDとして生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントID、上記サーバIDおよび上記第二のワンタイムIDを引数とする一方向関数 $h$ の関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記クライアントに対して送信する送信手段とを備えることを特徴とするサーバ。

39. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてサーバとの間で認証を行うクライアントであって、上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムIDとして生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバID、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムIDを暗号化し、この暗号化データと上記第一のワンタイムIDとを上記サーバに対して送信する送信手段と、

上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムIDとし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワンタイムID、上記第二の可変共有鍵、上記クライアントIDおよび上記サーバIDを引数とする一方向関数 $h$ の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記サーバから受信する受信手段と、

上記第二のワンタイムIDを演算により求め、この演算結果と、上記サーバか

ら受信した上記第二のワンタイムIDとの照合により、上記サーバを識別し、上記サーバを識別した場合に、上記サーバから受信した上記Diffie-Hellman公開値の他方と当該クライアントに予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数 $h$ の関数値を演算により求め、この演算結果と、上記サーバから受信した一方向関数 $h$ の関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするクライアント。

40. サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で認証を行い、

上記サーバは、上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムIDとして、この第一のワンタイムID、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバID、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワンタイムIDとを上記クライアントから受信する受信手段と、

上記第一のワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントID、上記サーバIDおよび上記第一のワンタイムIDに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムIDとして生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントID、上記サーバIDおよび上記第二のワンタイムIDを引数とする一方向関数 $h$ の関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記クライアントに対して送信する送信手段とを備え、また、

上記クライアントは、上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムIDとして生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバID、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムIDを暗号化し、この暗号化データと上記第一のワンタイムIDとを上記サーバに対して送信する送信手段と、

上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムIDとし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワンタイムID、上記第二の可変共有鍵、上記クライアントIDおよび上記サーバIDを引数とする一方向関数 $h$ の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記サーバから受信する受信手段と、

上記第二のワンタイムIDを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムIDとの照合により、上記サーバを識別し、上記サーバを識別した場合に、上記サーバから受信した上記Diffie-Hellman公開値の他方と当該クライアントに予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数 $h$ の関数値を演算により求め、この



演算結果と、上記サーバから受信した一方向関数 $h$ の関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とする認証システム。

41. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で相互に認証を行うサーバであって、

上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして、この第一のワンタイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして求め、この第二のワンタイムIDと上記第二の乱数とを上記クライアントに対して送信する送信手段と、

上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワンタイムIDとして、この第三のワンタイムIDを上記クライアントから受信する第二受信手段と、

上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするサーバ。

42. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイム

IDを用いてサーバとの間で相互に認証を行うクライアントであって、

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして、この第二のワンタイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムIDとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムIDとして求め、この第三のワンタイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とするクライアント。

43. サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で相互に認証を行うように構成され、

上記サーバは、上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして、この第一のワンタイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして求め、この第二のワンタイムIDと上記第二の乱数とを上記クライアントに対して送信する送信手段と、

上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワンタイムIDとして、この第三のワンタイムIDを上記クライアントから受信する第二受信手段と、

上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備え、また、

上記クライアントは、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして、この第二のワンタイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムIDとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムIDとして求め、この第三のワンタイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とする認証システム。

44. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはア

アプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で相互に認証を行うサーバであって、

上記クライアントとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして、この第一のワンタイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして求め、この第二のワンタイムIDと上記第二の乱数を上記クライアントに対して送信する送信手段と、

上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムIDとして、この第三のワンタイムIDを上記クライアントから受信する第二受信手段と、

上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするサーバ。

45. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてサーバとの間で相互に認証を行うクライアントであって、

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数を上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムIDとして、この第二のワнтаイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワнтаイムIDとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワнтаイムIDとして求め、この第三のワнтаイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とするクライアント。

46. サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワнтаイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワнтаイムIDを生成し、この生成されたワнтаイムIDを用いてクライアントとの間で相互に認証を行うように構成され、

上記サーバは、上記クライアントとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワнтаイムIDとして、この第一のワнтаイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムIDとして求め、この第二のワнтаイムIDと上記第二の乱数を上記クライアントに対して送信する送信手段と、

上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワнтаイムIDとして、この第三のワнтаイムIDを上記クライアントから受信する第二受信手段と、

上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワнтаイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワнтаイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備え、また、

上記クライアントは、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワнтаイムIDとして求め、この第一のワнтаイムIDと上記第一の乱数を上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムIDとして、この第二のワнтаイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワнтаイムIDとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワнтаイムIDとして求め、この第三のワнтаイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とする認証システム。

47. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワнтаイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワнтаイムIDを生成し、この生成されたワнтаイムIDを用いてクライアントとの間で相互に認証を行うサーバであって、

上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワнтаイムIDとして、この第一のワнтаイムIDを上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライア

ントID、当該サーバに予め設定されたサーバIDを上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、

上記第一のワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアントIDおよび上記サーバIDに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムIDとして求め、上記クライアントID、上記サーバIDおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムIDとを上記クライアントに対して送信する送信手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするサーバ。

48. 複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてサーバとの間で相互に認証を行うクライアントであって、

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムIDとを上記サーバに対して送信する送信手段と、

上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムIDとして、この第二のワンタイムIDを上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアントIDおよび上記サーバIDを上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、

上記第二のワンタイムIDを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムIDとの照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバIDおよび上記クライアントIDに基づいて、上記サーバの正当性を判定する判定手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするクライアント。

49. サーバとクライアントとにより構成された認証システムにおいて、

上記サーバおよびクライアントは、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成し、この生成されたワンタイムIDを用いてクライアントとの間で相互に認証を行うように構成され、

上記サーバは、上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムIDとして、この第一のワンタイムIDを上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバIDを上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、

上記第一のワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムIDとの照合により、上記クライアント



を識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアント ID および上記サーバ ID に基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイム ID として求め、上記クライアント ID、上記サーバ ID および上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイム ID とを上記クライアントに対して送信する送信手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備え、また、

上記クライアントは、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイム ID として求め、当該クライアントに予め設定されたクライアント ID、上記サーバに予め設定されたサーバ ID および上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイム ID とを上記サーバに対して送信する送信手段と、

上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイム ID として、この第二のワンタイム ID を上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアント ID および上記サーバ ID を上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、

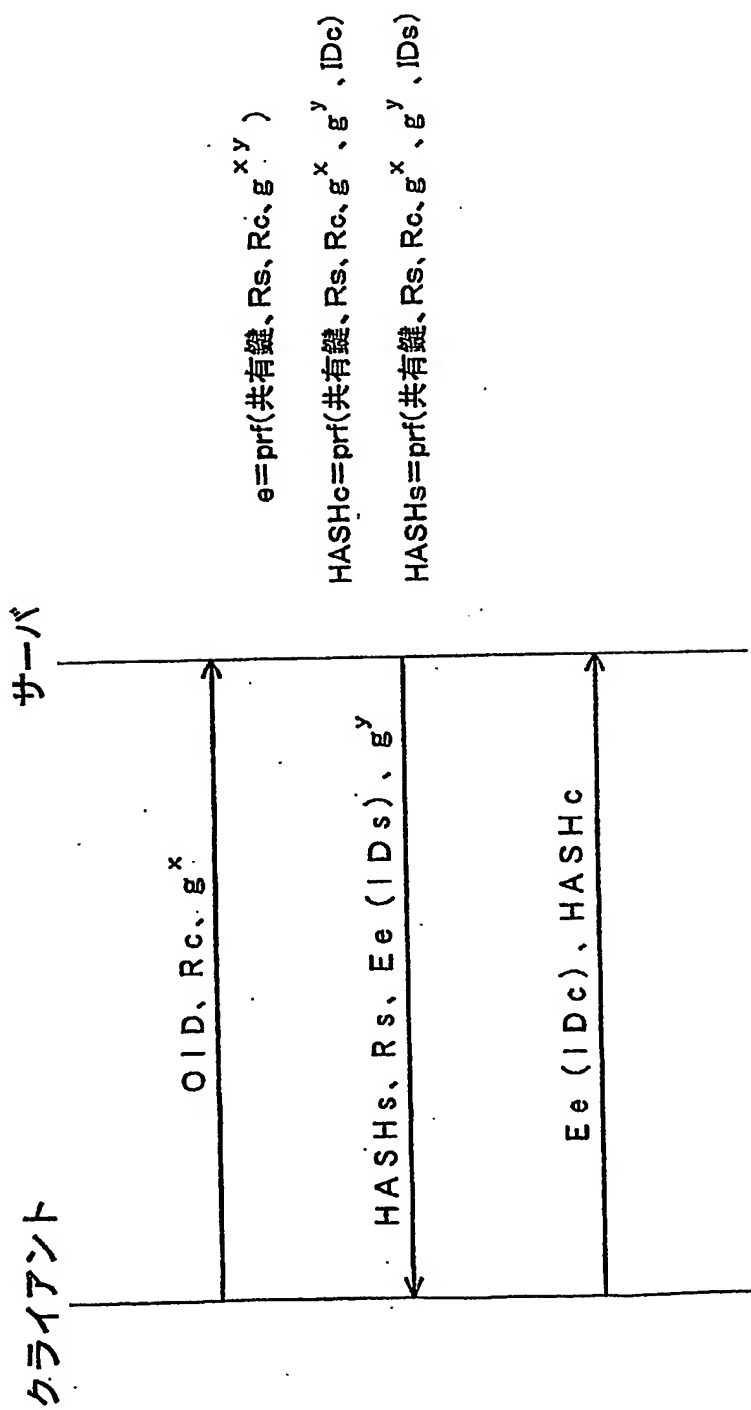
上記第二のワンタイム ID を演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイム ID との照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバ ID および上記クライアント ID に基づいて、上記サーバの正当性を判定する判定手段と、

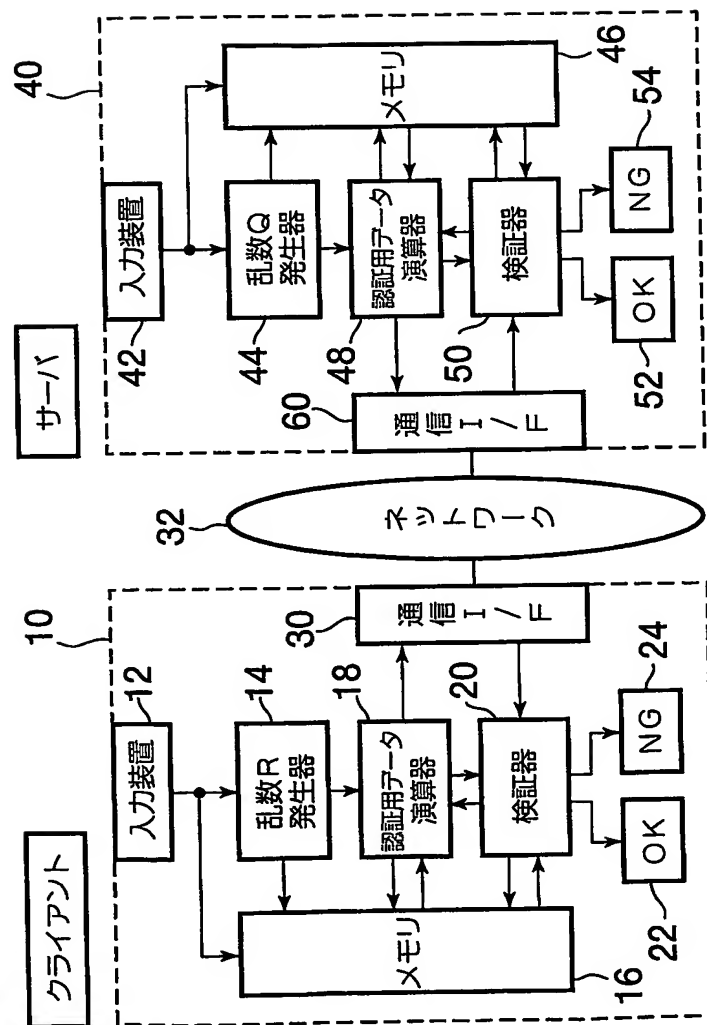
上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とする認証システム。

50. 上記サーバおよび上記クライアントは、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後で、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようになっていることを特徴とする請求項49に記載の認証システム。

1 / 1 6

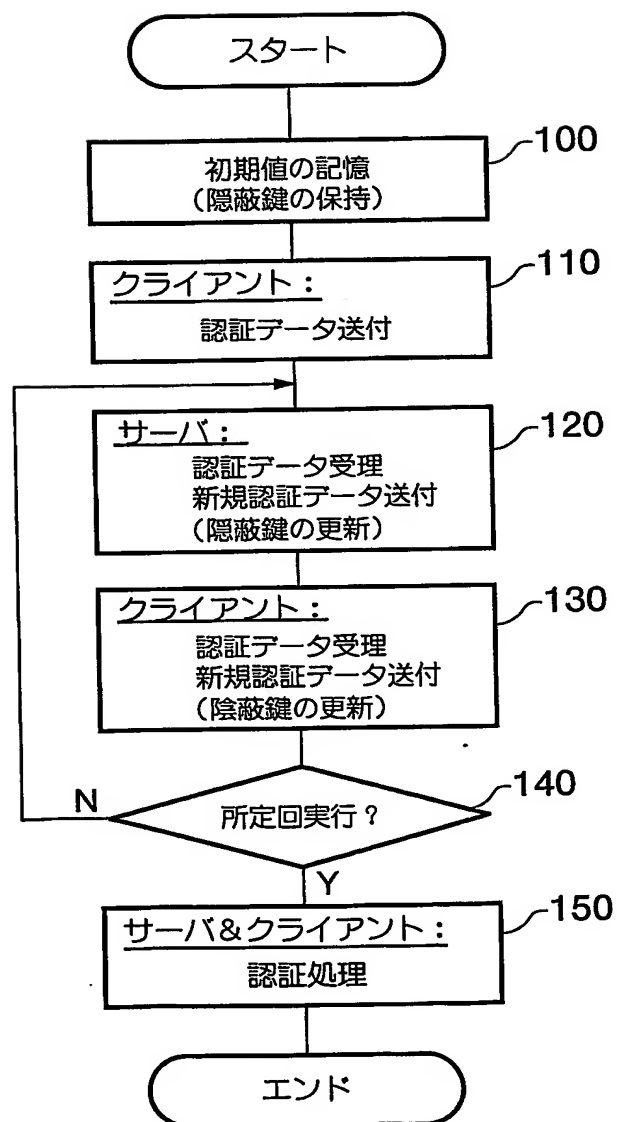
図 1





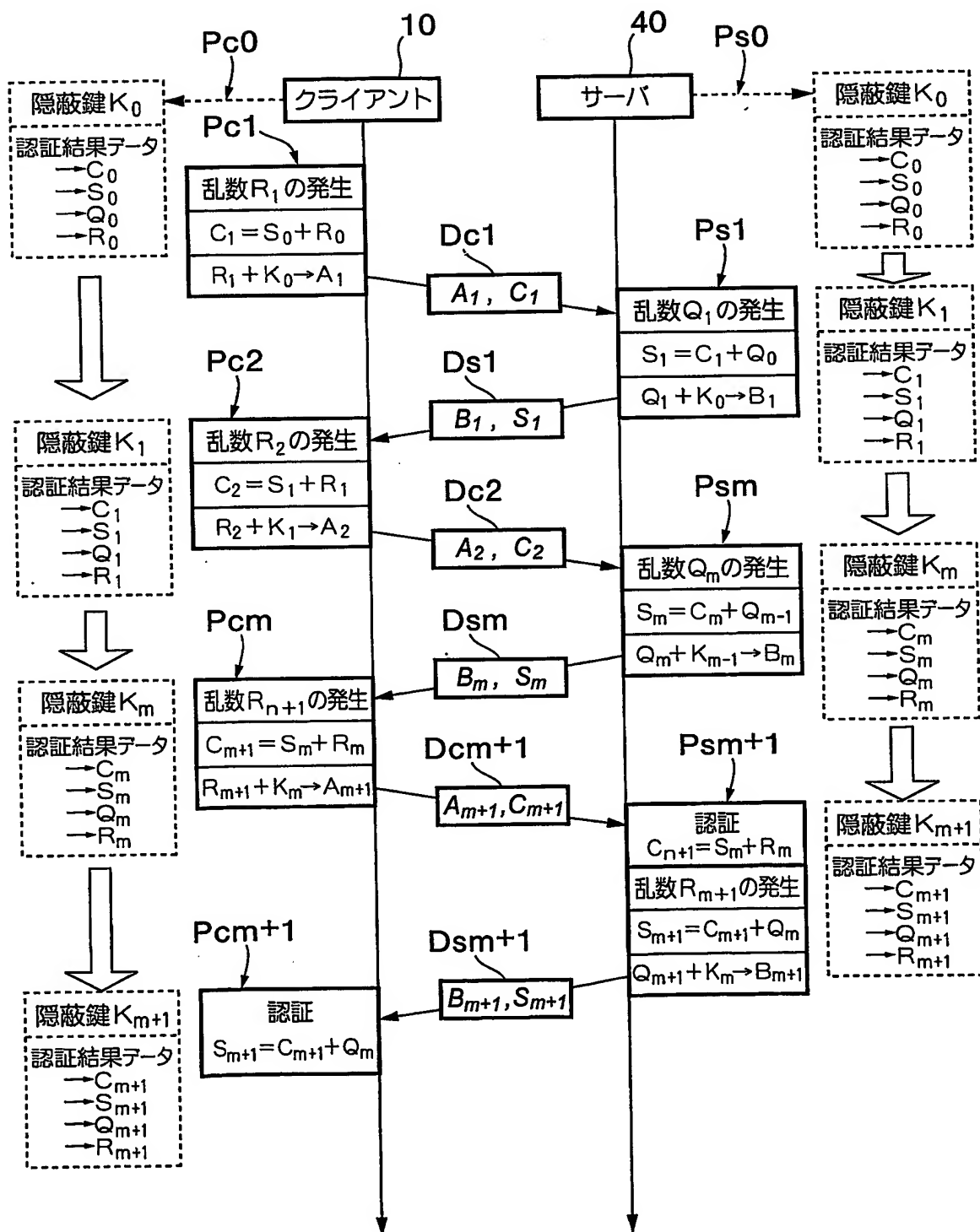
3 / 16

図 3



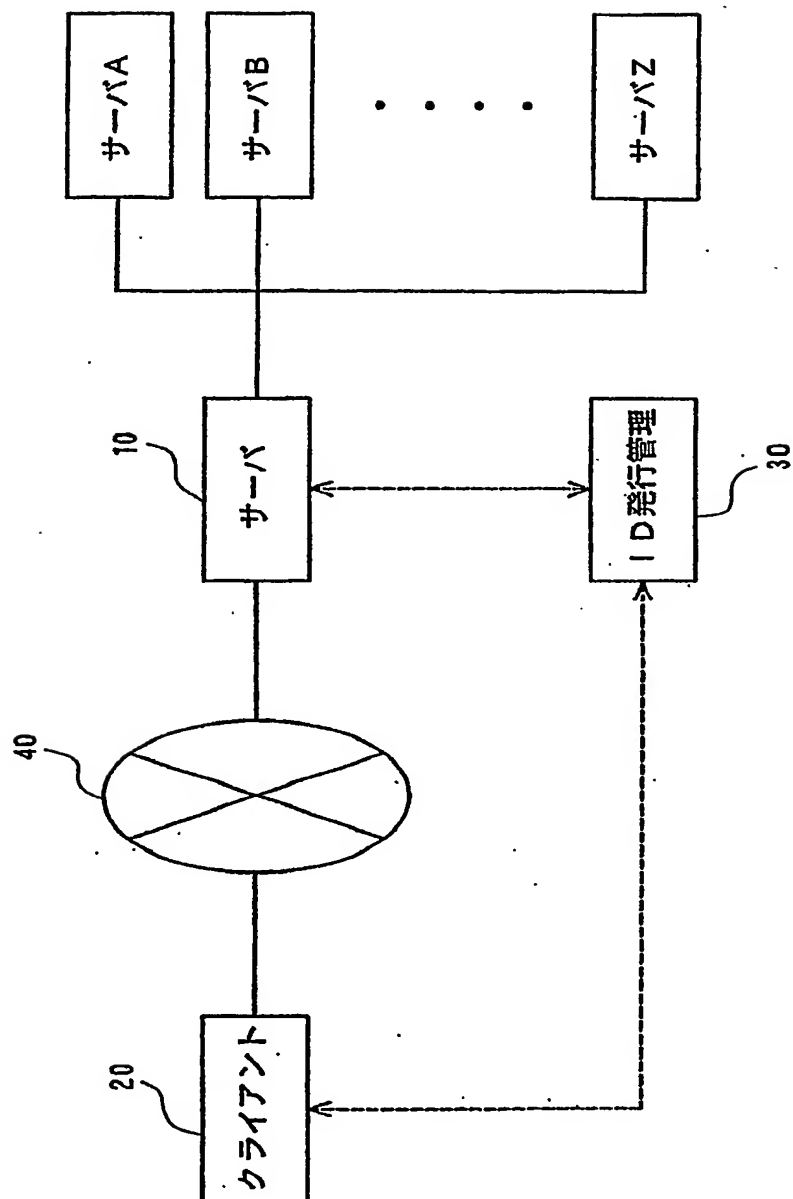
4 / 16

図 4



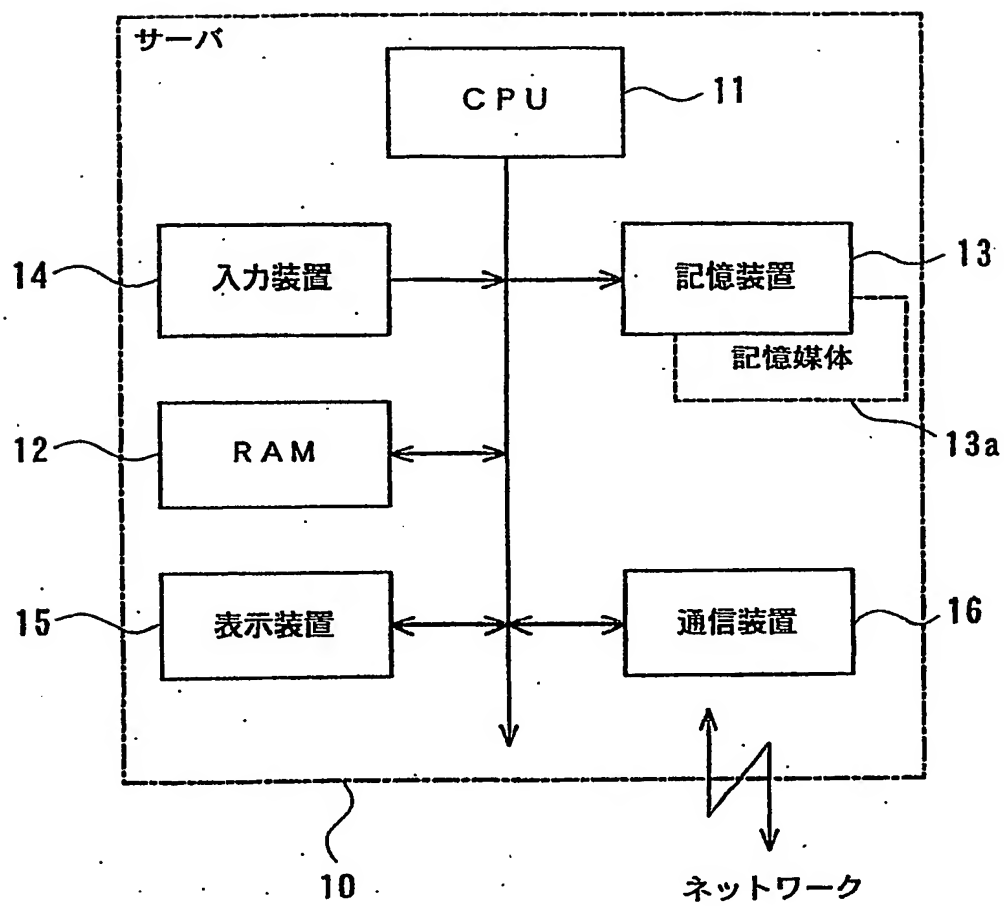
5 / 16

図 5



6 / 16

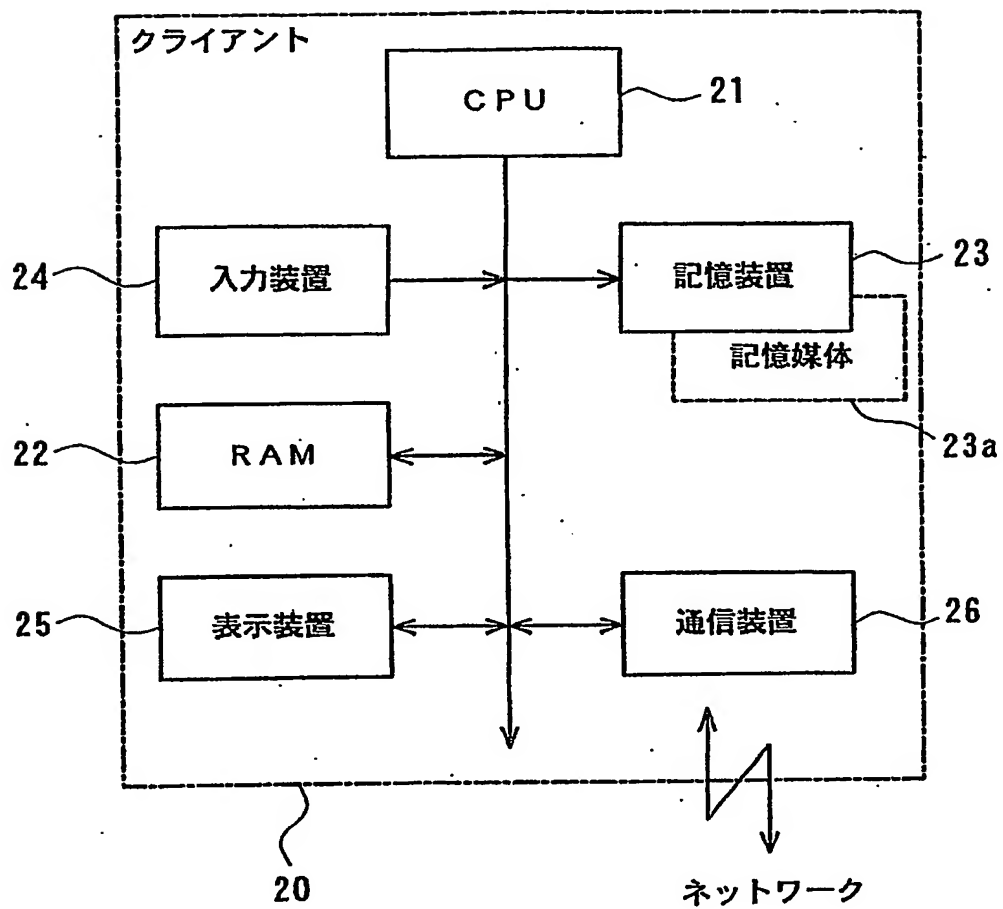
図 6

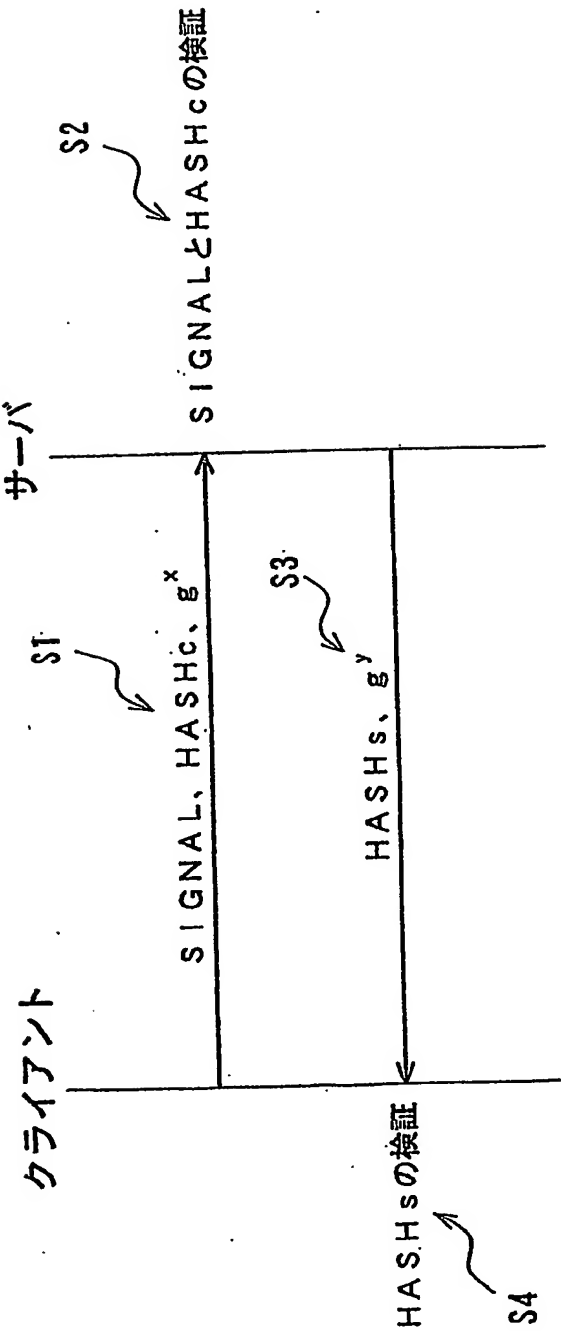


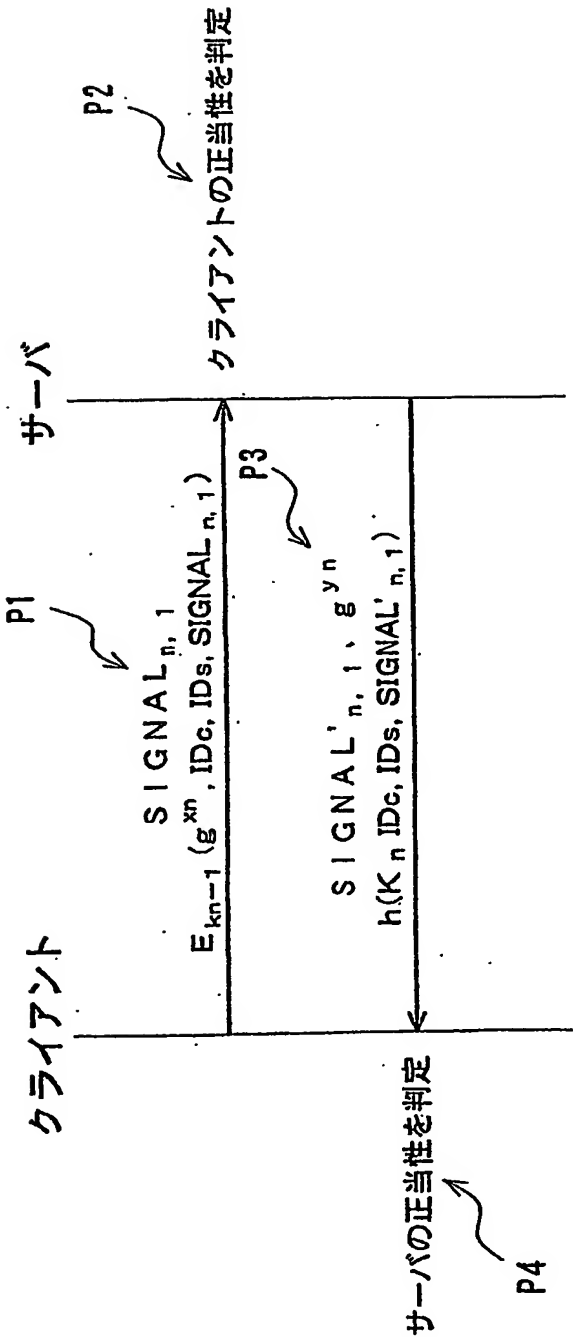


7 / 16

図 7

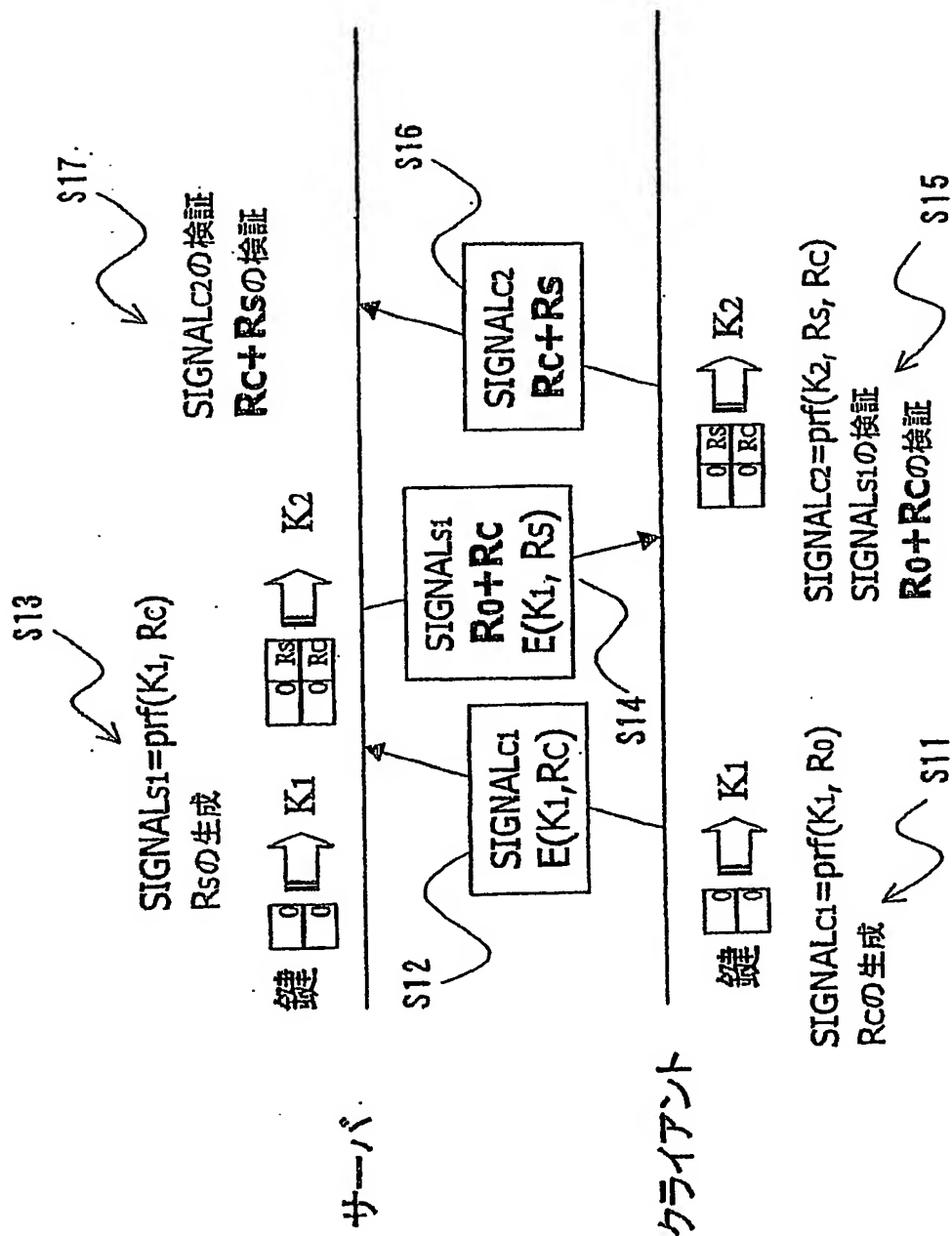






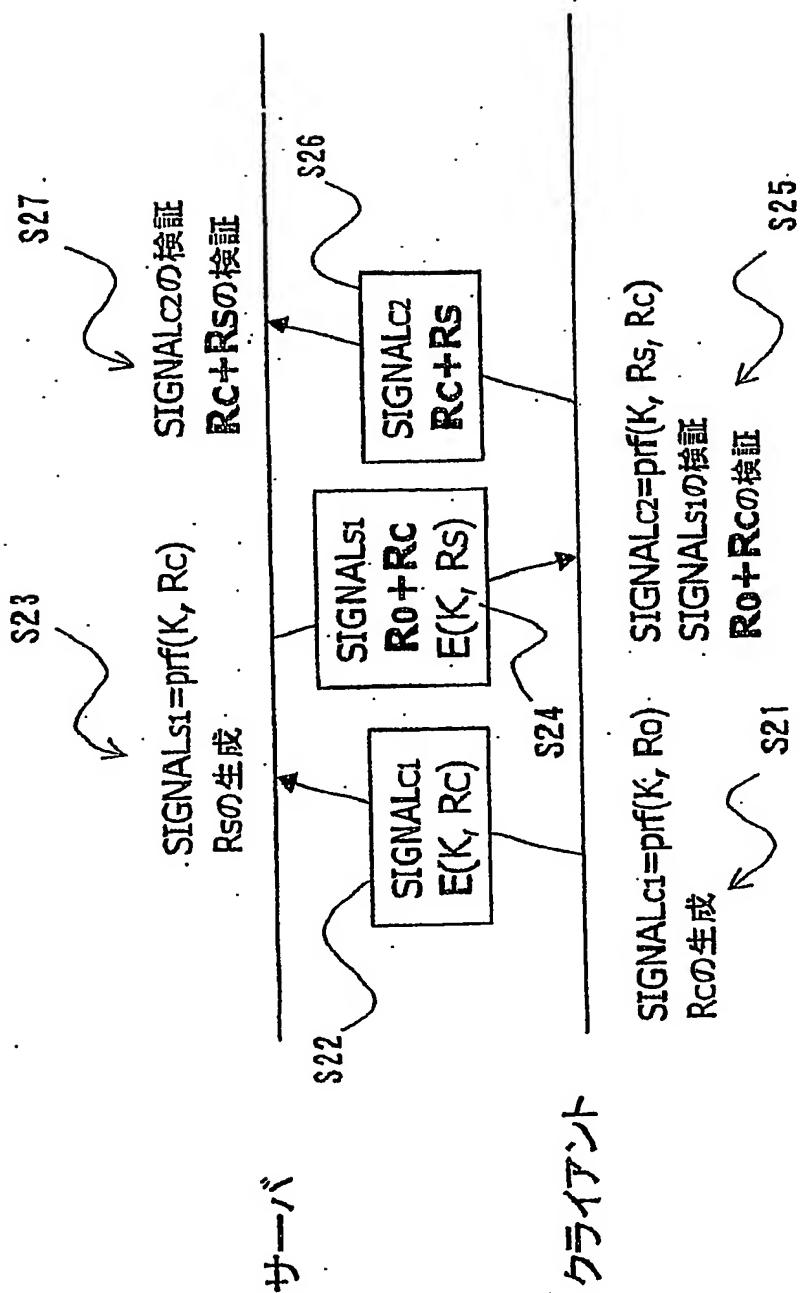
10 / 16

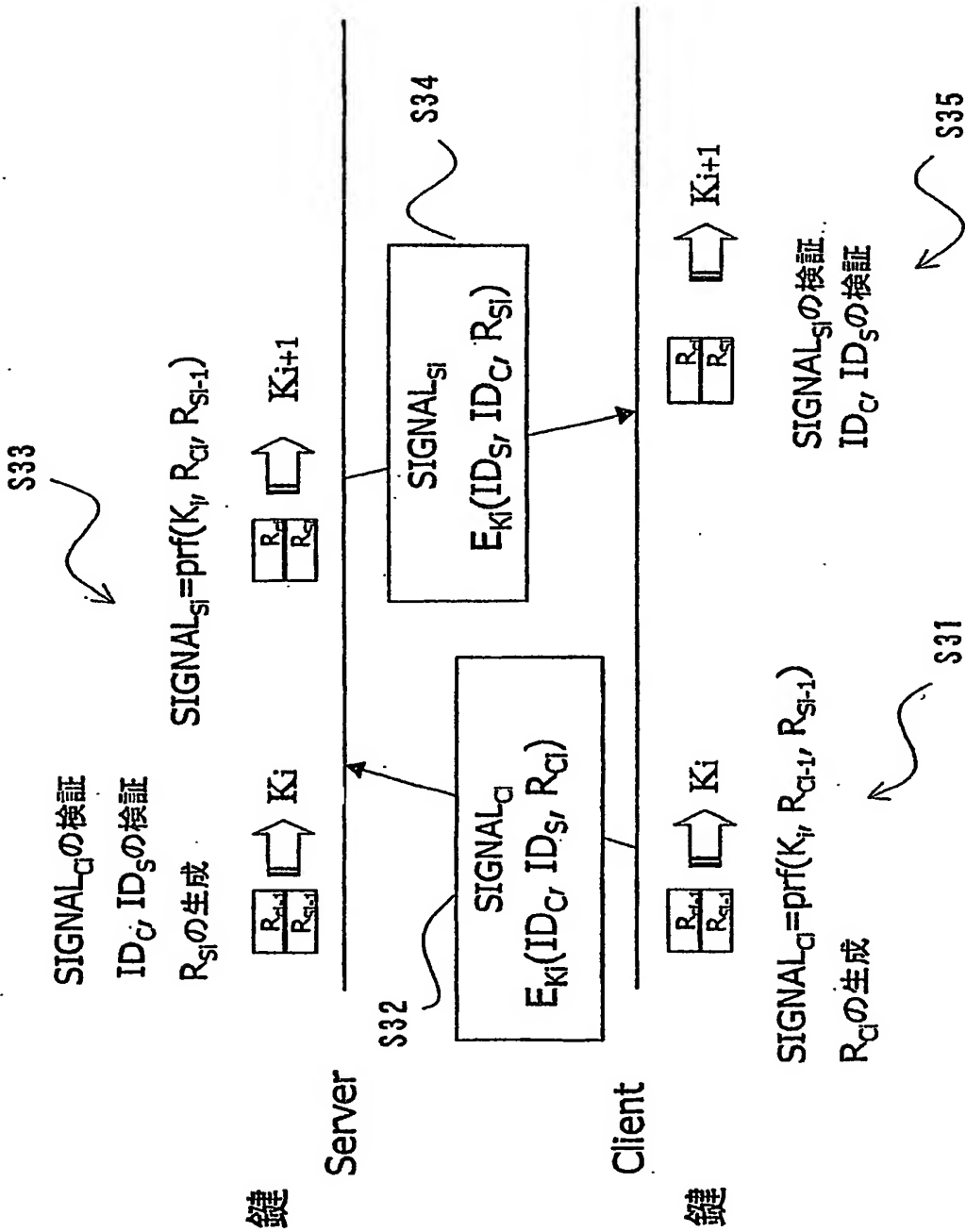
図 10

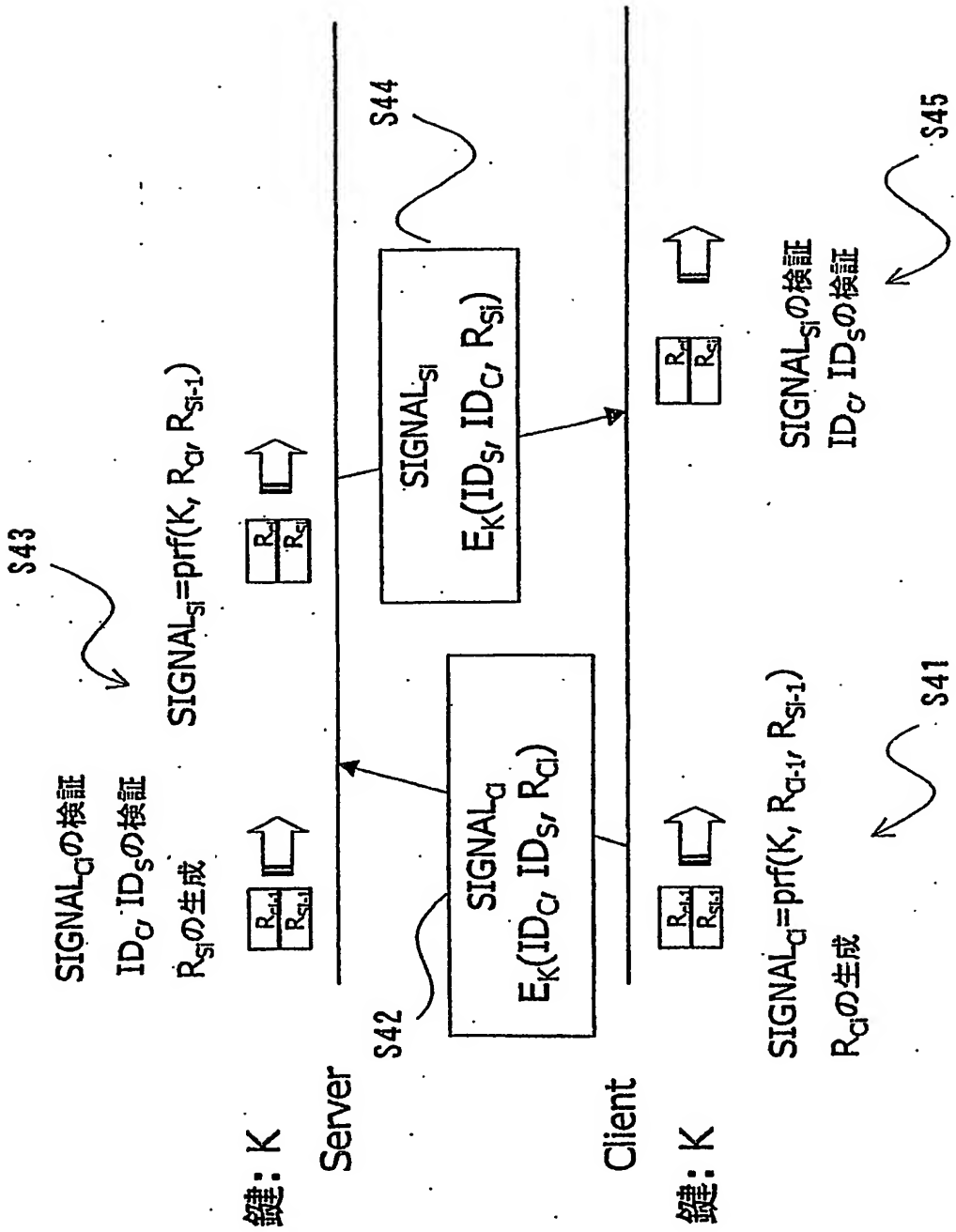


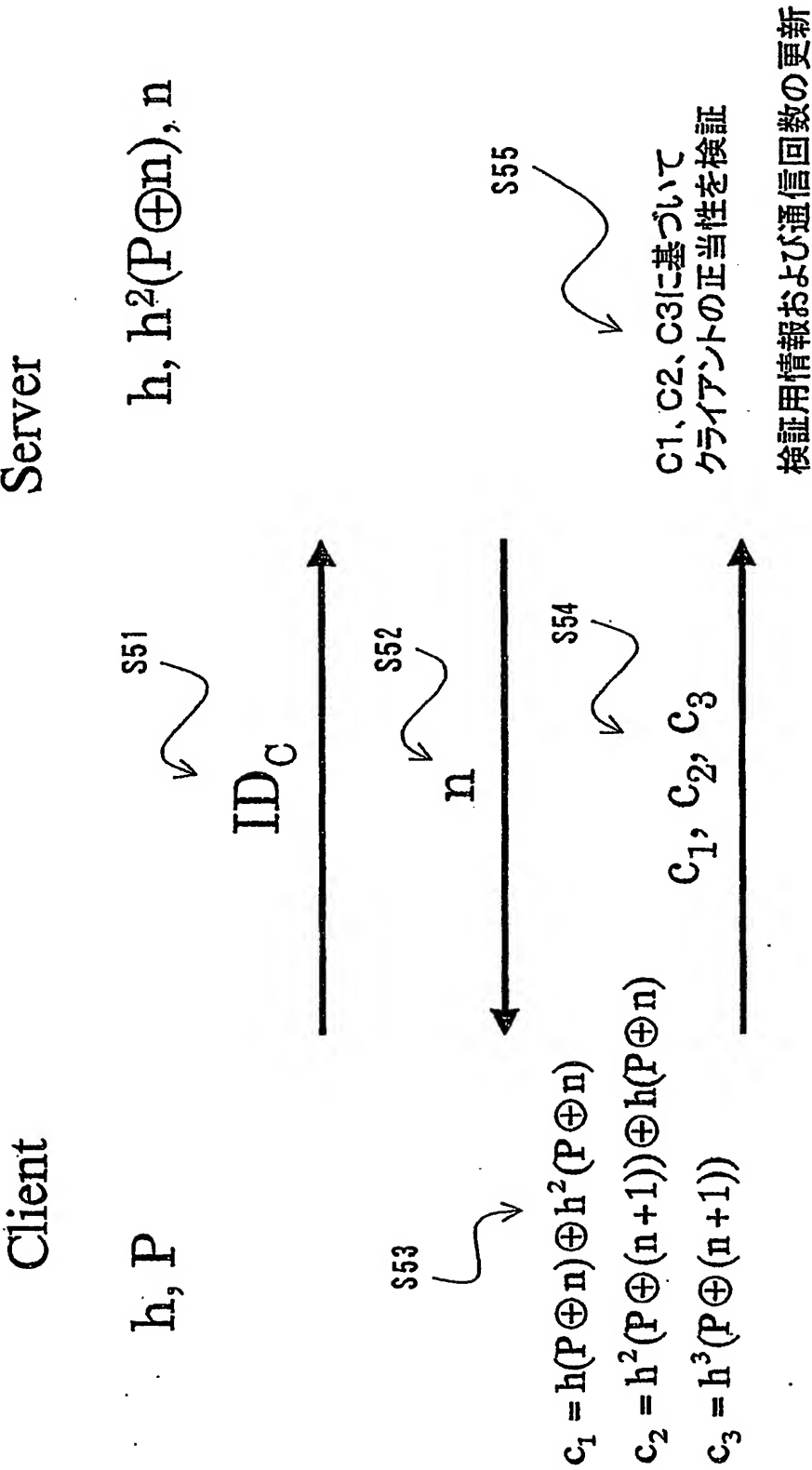
1 1 / 1 6

図 1 1

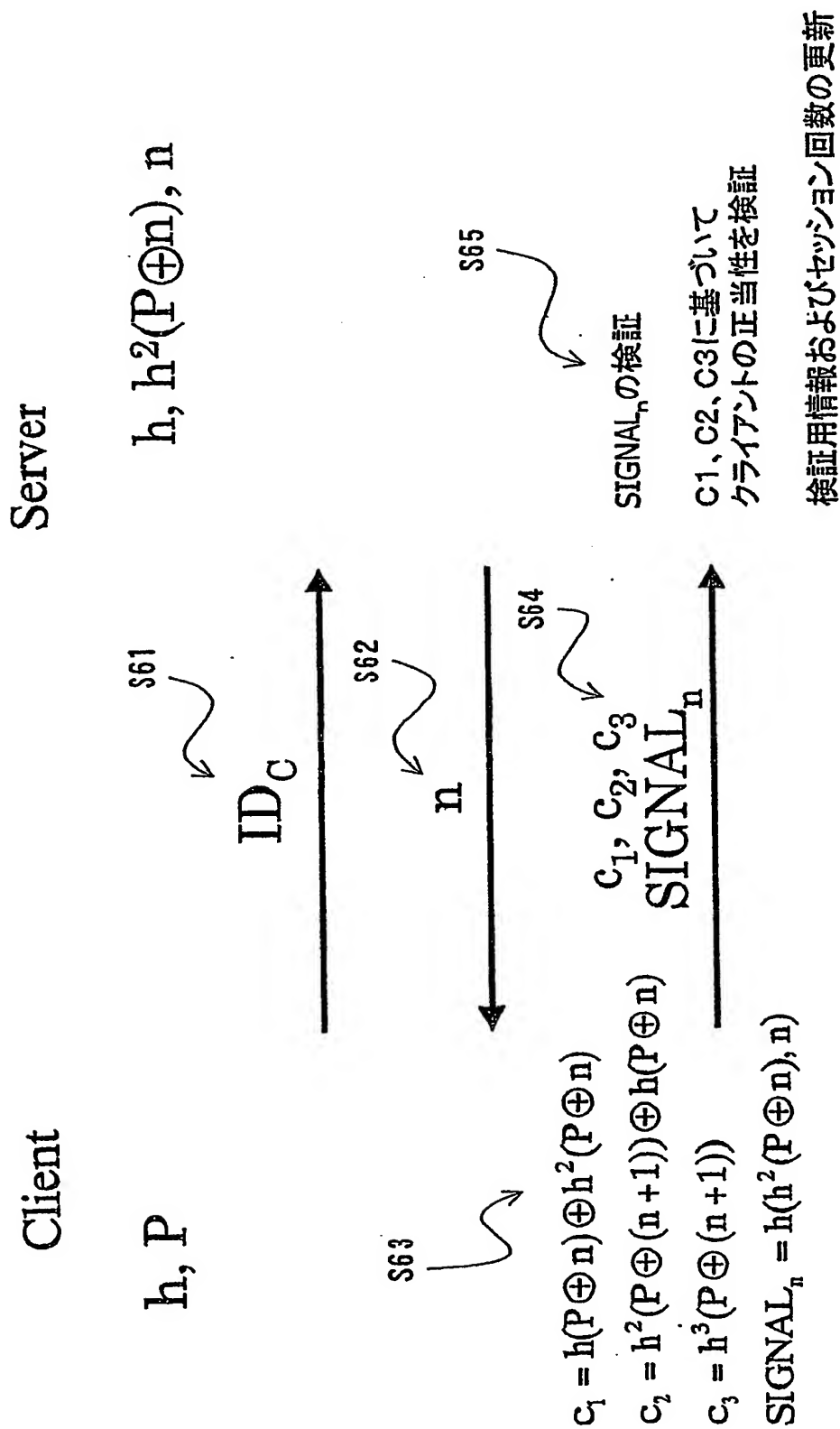


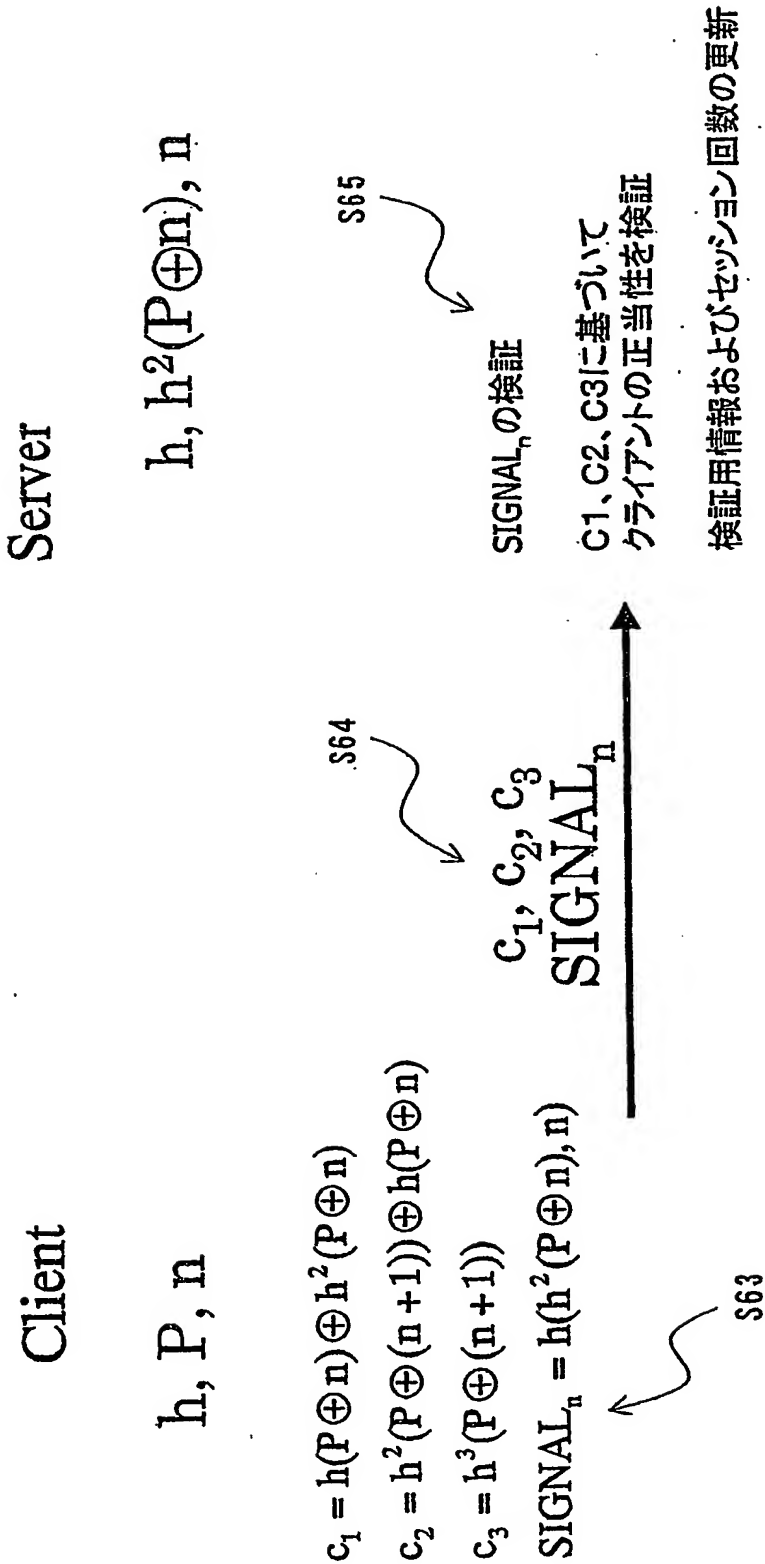












# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/07794

## A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl<sup>7</sup> H04L9/32

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl<sup>7</sup> H04L9/32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Toroku Jitsuyo Shinan Koho	1994-2003
Kokai Jitsuyo Shinan Koho	1971-2003	Jitsuyo Shinan Toroku Koho	1996-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

JICST FILE(JOIS), WPI, one-time, identity

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HANDBOOK of APPLIED CRYPTOGRAPHY, CRC Press, 1997, pages 400 to 403, especially (ii) Challenge-response based on(keyed) one-way functions	16, 19, 25, 44-46
A	Smafati, D. & Molva, R., A method providing identity privacy to mobile users during authentication, Proceedings of Workshop on Mobile Computing Systems and Applications, 1995, pages 196 to 199, especially 4,2 Alias Computation	14-26, 28-50

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

\* Special categories of cited documents:  
 "A" document defining the general state of the art which is not considered to be of particular relevance  
 "E" earlier document but published on or after the international filing date  
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)  
 "O" document referring to an oral disclosure, use, exhibition or other means  
 "P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention  
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone  
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art  
 "&" document member of the same patent family

Date of the actual completion of the international search  
 21 November, 2003 (21.11.03)

Date of mailing of the international search report  
 09 December, 2003 (09.12.03)

Name and mailing address of the ISA/  
 Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

# INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP03/07794

## Box I Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☒ Claims Nos.: 1-13, 27  
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:  
See extra sheet for claims 1-13.  
As for claim 27 (authentication method), it is not described to which claim it refers.
3. ☐ Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

## Box II Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

The inventions of claims 1-13 relate to inter-authentication not related to the one time ID. The inventions of claims 14-26, 28-50 relate to the one time ID. It should be noted that since the inter-authentication is a known technique and cannot be a "special technical feature".

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest ☐ The additional search fees were accompanied by the applicant's protest.  
☐ No protest accompanied the payment of additional search fees.

**INTERNATIONAL SEARCH REPORT**

International application No.

PCT/JP03/07794

Continuation of Box No.I-2 of continuation of first sheet(1)

In claim 1, it is unknown whether the "new storage data generated" in the description "the new storage data generated is encrypted by using the history data and transmitted to the second authentication device" is the same data as the "storage data" in the description "the first authentication device generates new storage data by using the history data stored". Similarly, it is unknown whether the "new storage data generated" in the description "the new storage data generated is encrypted by using the history data and transmitted to the first authentication device" is the same data as the "storage data" in the description "the second authentication device generates new storage data by using the history data stored". Moreover, if they are different data, it is unknown what kind of meaning is present in generating storage data by using the history data. Similarly, it is unknown whether the "storage data from the first authentication device" in the description "the second authentication device generates new storage data by using storage data from the first authentication device" is the same data as the "storage data" in the description "first transmission step for encrypting the new storage data generated and transmitting it to the second authentication device". If they are different data, it is unknown what kind of data is the "storage data from the first authentication device".

Furthermore, there are descriptions that "the first authentication device generates new storage data by using the history data stored" and "the second authentication device generates new storage data by using the history data stored". In each of these descriptions, it is unknown whether the data is the same data as the "history data" in the description "using as the history data the updated result which has been updated by using the storage data by the previous authentication." If they are different data, it is unknown what kind of data is "history data" in these descriptions and "said history data" in the subsequent description.

Moreover, it is unknown what kind of data is the "storage data by the previous authentication" in the description "using as the history data the updated result which has been updated by using the storage data by the previous authentication".

That is, it is unknown how the "inter-authentication method" is performed because the relationship between the plurality of "storage data" and "history data" is unclear. The same applies to claims 2-13 which refer to claim 1.

## A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl<sup>7</sup> H04L9/32

## B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl<sup>7</sup> H04L9/32

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2003年
日本国登録実用新案公報	1994-2003年
日本国実用新案登録公報	1996-2003年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

JICSTファイル (JOIS), WPI  
one-time, identity

## C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	HANDBOOK of APPLIED CRYPTOGRAPHY, CRC Press, 1997, p.400-403 especially (ii) Challenge-response based on(keyed)one-way functions	16, 19, 25, 44-46
A	Samfat, D. & Molva, R., A method providing identity privacy to mobile users during authentication, Proceedings of Workshop on Mobile Computing Systems and Applications, 1995, p.196-199, especially 4,2 Alias Computation	14-26, 28-50

☐ C欄の続きにも文献が列举されている。☐ パテントファミリーに関する別紙を参照。

## \* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技术水準を示すもの  
「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの  
「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)  
「O」 口頭による開示、使用、展示等に言及する文献  
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献  
「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの  
「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの  
「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの  
「&」 同一パテントファミリー文献

国際調査を完了した日

21. 11. 03

国際調査報告の発送日

09.12.03

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)  
郵便番号100-8915  
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)  
中里 裕正



5M 9364

電話番号 03-3581-1101 内線 3597

## 第 I 欄 請求の範囲の一部の調査ができないときの意見 (第 1 ページの 2 の続き)

法第 8 条第 3 項 (PCT 17 条 (2) (a)) の規定により、この国際調査報告は次の理由により請求の範囲の一部について作成しなかった。

1. ☐ 請求の範囲 \_\_\_\_\_ は、この国際調査機関が調査をすることを要しない対象に係るものである。  
つまり、
2. ☒ 請求の範囲 1-13, 27 は、有意義な国際調査をすることができる程度まで所定の要件を満たしていない国際出願の部分に係るものである。つまり、  
請求の範囲 1-13 については、別紙を参照されたい。  
請求の範囲 27 には、「請求項に記載の認証方法」と記載されており、いずれの請求項を引用しているのか不明である。
3. ☐ 請求の範囲 \_\_\_\_\_ は、従属請求の範囲であって PCT 規則 6.4(a) の第 2 文及び第 3 文の規定に従って記載されていない。

## 第 II 欄 発明の単一性が欠如しているときの意見 (第 1 ページの 3 の続き)

次に述べるようにこの国際出願に二以上の発明があるところの国際調査機関は認めた。

請求の範囲 1-13 は、ワンタイム ID とは何ら関係しない相互認証に関する発明であり、請求の範囲 14-26, 28-50 は、ワンタイム ID に関する発明である。なお、相互認証は周知の技術であるから、相互認証を「特別な技術的特徴」とすることはできない。

1. ☐ 出願人が必要な追加調査手数料をすべて期間内に納付したので、この国際調査報告は、すべての調査可能な請求の範囲について作成した。
2. ☒ 追加調査手数料を要求するまでもなく、すべての調査可能な請求の範囲について調査することができたので、追加調査手数料の納付を求めなかった。
3. ☐ 出願人が必要な追加調査手数料を一部のみしか期間内に納付しなかったため、この国際調査報告は、手数料の納付のあった次の請求の範囲のみについて作成した。
4. ☐ 出願人が必要な追加調査手数料を期間内に納付しなかったため、この国際調査報告は、請求の範囲の最初に記載されている発明に係る次の請求の範囲について作成した。

## 追加調査手数料の異議の申立てに関する注意

- ☐ 追加調査手数料の納付と共に出願人から異議申立てがあった。  
☐ 追加調査手数料の納付と共に出願人から異議申立てがなかった。

請求の範囲1の「生成した新規の記憶データを前記履歴データを用いて暗号化して第2認証装置に送信する」との記載における「生成した新規の記憶データ」が、「前記第1認証装置は、記憶されている履歴データを用いて記憶データを新規に生成し」との記載における「記憶データ」と同一のデータであるのか否か不明であり、同様に、「生成した新規の記憶データを前記履歴データを用いて暗号化して第1認証装置に送信する」との記載における「生成した新規の記憶データ」が、「前記第2認証装置は、…記憶されている履歴データを用いて記憶データを新規に生成し」との記載における「記憶データ」と同一のデータであるのか否かという点も不明である。また、それぞれ別個のデータであるならば、履歴データを用いて記憶データを生成することによどのような意味があるのか不明である。同じく、「前記第2認証装置は、前記第1認証装置からの記憶データ…を用いて新規に記憶データを生成し」との記載における「第1認証装置からの記憶データ」とは、「生成した新規の記憶データを…暗号化して第2の認証装置に送信する第1送信工程」との記載における「記憶データ」と同一のデータであるのか否か不明であり、異なるデータであるならば、「第1認証装置からの記憶データ」とは、如何なるデータであるのか不明である。

さらに「前記第1認証装置は、記憶されている履歴データを用いて記憶データを新規に生成し」及び「前記第2認証装置は、…記憶されている履歴データを用いて記憶データを新規に生成し」と記載されているが、各々、「前回の認証による記憶データを用いて更新した更新結果を履歴データとして」との記載における「履歴データ」と同一のデータであるのか不明であり、異なるデータであるならば、これらの記載における「履歴データ」及びその後の記載における「前記履歴データ」とは、如何なるデータであるのか不明である。

また「前回の認証による記憶データを用いて更新した更新結果を履歴データとして」との記載における「前回の認証による記憶データ」とは如何なるものかということも不明である。

すなわち請求の範囲1に記載された「相互認証方法」については、複数存在する「記憶データ」及び「履歴データ」の間の関係が不明であるため、どのような手順にて認証が行われているのか把握することができない。請求の範囲1を引用する請求の範囲2～13についても同様である。